Enhance Salesforce with Point and Click Tools

Salesforce, Winter '17

@salesforcedocs

Last updated: December 21, 2016

CONTENTS

WELCOME, SALESFORCE POINT & CLICK ADMINISTRATORS

Ready to go beyond the basics of Salesforce administration? Want to customize your organization, push its boundaries, and enhance its functionality? You can do that and so much more without writing a single line of code. All you need is your mouse and a sense of adventure. Put on your fedora, it's time to explore!

Inside you'll find a multitude of features you can use to expand your organization. Enhance your objects, data, and fields, customize your organization's look and feel, augment your business processes, create websites, and even create apps—all using point-and-click tools.

Find Object Management Settings

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which experience of Salesforce you have enabled, these customizations are located in different areas of Setup.

IN THIS SECTION:

Find Object Management Settings in Lightning Experience

Salesforce lets you customize your object model with features like custom fields, page layouts, and validation rules. Most objects are available from the Object Manager in Setup.

EDITIONS

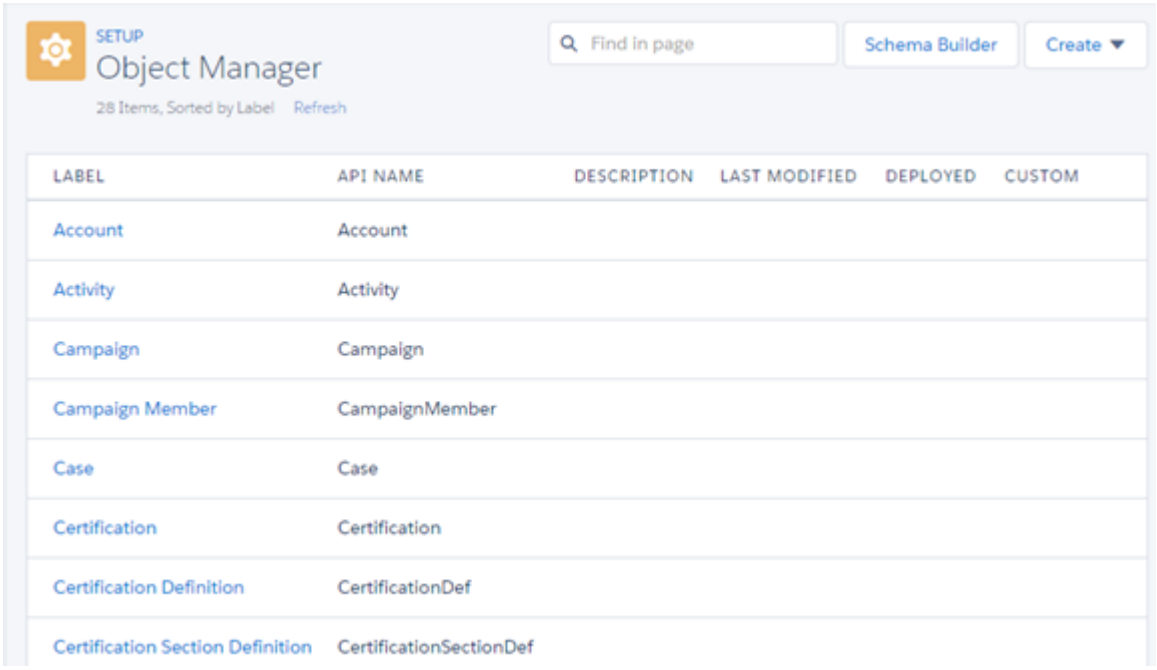
Available in: Lightning Experience

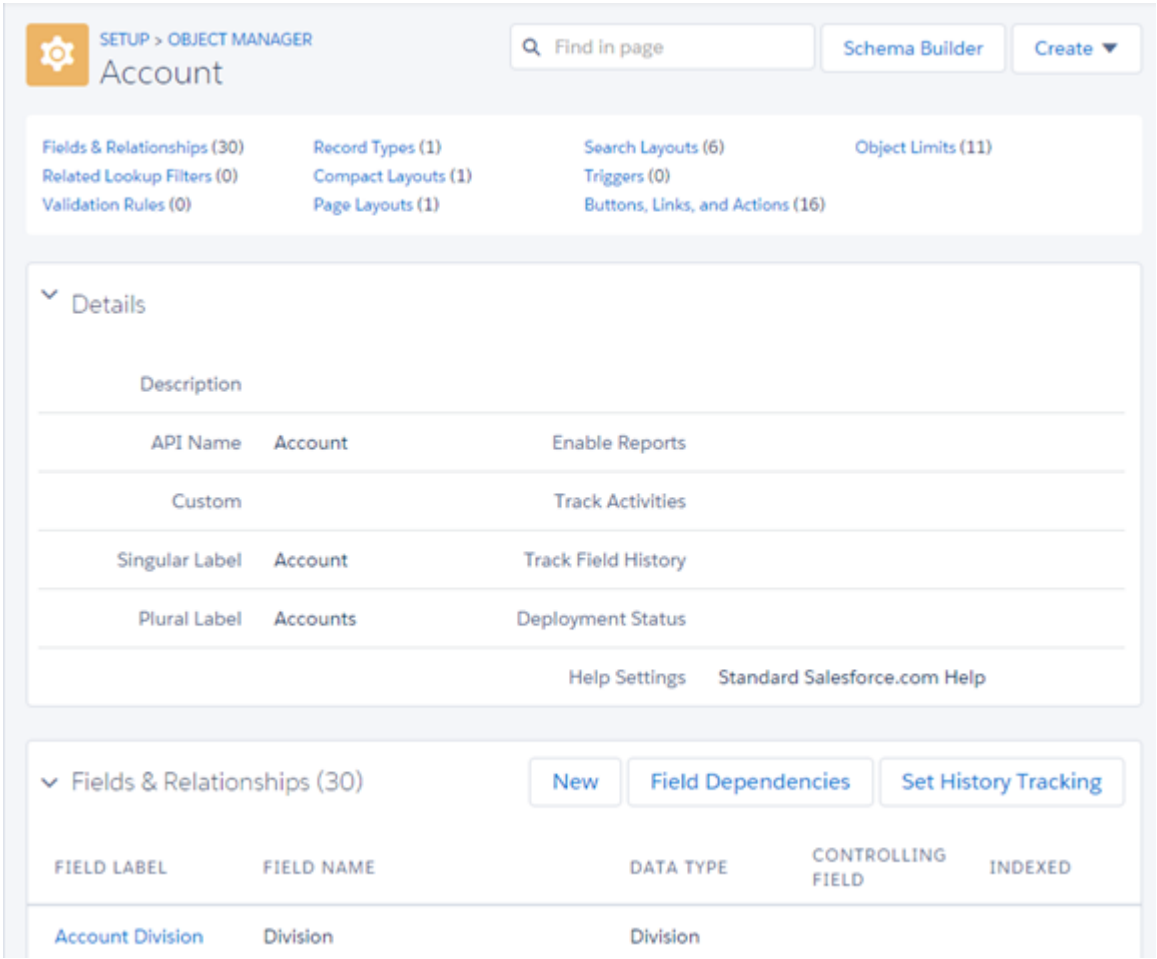
Available in all editions

Standard Objects and Custom Objects

A *standard object*, such as Account or Contact, comes out of the box with your Salesforce organization. A *custom object* is an object that you or another administrator created.

From Setup, enter `Object Manager` in the `Quick Find` box, then select **Object Manager**. Select one of the objects in the list, and then scroll to the section for the specific customization.


SETUP
Object Manager
28 Items, Sorted by Label Refresh
Find in page
Schema Builder
Create
LABEL
API NAME
DESCRIPTION
LAST MODIFIED
DEPLOYED
CUSTOM
Account
Account
Activity
Activity
Campaign
Campaign
Campaign Member
CampaignMember
Case
Case
Certification
Certification
Certification Definition
CertificationDef
Certification Section Definition
CertificationSectionDef


SETUP > OBJECT MANAGER
Account
Find in page
Schema Builder
Create
Fields & Relationships (30)
Related Lookup Filters (0)
Validation Rules (0)
Record Types (1)
Compact Layouts (1)
Page Layouts (1)
Search Layouts (6)
Triggers (0)
Buttons, Links, and Actions (16)
Object Limits (11)
Details
Description
API Name
Account
Custom
Singular Label
Account
Plural Label
Accounts
Enable Reports
Track Activities
Track Field History
Deployment Status
Help Settings
Standard Salesforce.com Help
Fields & Relationships (30)
New
Field Dependencies
Set History Tracking
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Account Division
Division
Division

For example, to add a custom field to the Account object, enter *Object Manager* in the Quick Find box, then select **Object Manager**. Next, select **Account**, and then scroll to Fields & Relationships.

Other Standard Objects

Some standard objects aren't housed in the Object Manager. To access customization settings for one of those objects, from Setup, enter the object name in the Quick Find box, then select the customization setting.

From Setup, enter the object name in the Quick Find box, then select the customization.

For example, to add a trigger to the Groups object, enter *Group* in the Quick Find box, then select **Group Triggers**.

External Objects

An *external* object is similar to custom objects, except that it maps to data that's stored outside your Salesforce organization.

From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**. Next, click one of the external objects in the list. Then scroll to the section for the specific customization.

For example, to add a custom field to the Orders external object, enter *External Objects* in the Quick Find box, then select **External Objects**. Click **Orders**, and then scroll to Custom Fields and Relationships.

Find Object Management Settings in Salesforce Classic

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which type of object you want to find, these customizations are located in different areas of Setup.

EDITIONS

Available in: Salesforce Classic

Available in all editions

Standard Objects

A *standard object*, such as Account or Contact, comes out of the box with your Salesforce organization.

From Setup, enter the name of the appropriate object in the Quick Find box, then select the specific customization.

For example, to add a custom field to the Case object, enter *Case* in the Quick Find box, then select **Fields** under Cases.

Custom Objects

A *custom object* is an object that you or another administrator created.

From Setup, enter *Objects* in the Quick Find box and select **Objects**. Next, click one of the custom objects in the list. Then scroll to the section for the specific customization.

For example, to add a custom field to the Job Applications object, enter *Objects* in the Quick Find box, then select **Objects**. Click **Job Applications**, and then scroll to Custom Fields and Relationships.

External Objects

An *external* object is similar to custom objects, except that it maps to data that's stored outside your Salesforce organization.

From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**. Next, click one of the external objects in the list. The scroll to the section for the specific customization.

For example, to add a custom field to the Orders external object, enter *External Objects* in the Quick Find box, then select **External Objects**. Click **Orders**, and then scroll to Custom Fields and Relationships.

Customize Your Salesforce Org

Watch a Demo: How to Change the Look and Feel of Salesforce for Your Company

Quick demo of how to customize the way Salesforce looks for your organization.

You can customize each of the standard tabs and types of records, including adding custom fields and setting page layouts. You can also customize search, tagging, and user interface options for your org. In addition, every Contact Manager, Group, Professional, Enterprise, Unlimited, and Performance Edition user can customize various personal display options.

To tailor Salesforce for your org, you can customize the display of the various tabs and other items. Select a link to get started on any task.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available customization options vary according to which Salesforce Edition you have.

USER PERMISSIONS

To view setup options:
- "View Setup and Configuration"

To customize your org:
- "Customize Application"

Force.com Quick Access Menu

The Force.com quick access menu offers handy shortcuts to customization features.

When you're working on apps or objects, use this menu to jump to relevant app customization features. It's available from object list view pages and record detail pages.

Note: If drag-and-drop scheduling on list views is enabled, the Force.com quick access menu isn't visible for list views on accounts, contacts, and custom objects.

- To expand or collapse the menu, click (or press ALT+;).
- To scroll down the list of the menu, press TAB.
- To select an option on the menu, press ENTER.
- To remove the menu from all list views and record pages, click **Turn off menu**.

To restore the quick access menu:

1. From your personal settings, enter `Advanced User Details` in the `Quick Find` box, then select **Advanced User Details**. No results? Enter `Personal Information` in the `Quick Find` box, then select **Personal Information**.
2. Click **Edit**.
3. Select the **Force.com Quick Access Menu** checkbox.
4. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view the Force.com quick access menu:
- "Customize Application"

Rename Object, Tab, and Field Labels

You can change the name of almost any object, field, or tab in Salesforce. This simple adjustment lets you continue using the terminology your users already know and helps them transition to using Salesforce. However, Salesforce Help and most pages in Setup always display the original names for standard objects, fields, and tabs.

For example, you can change the name label for the "Accounts" object and related "Accounts" tab to "Companies", and change the field "Account Name" to "Company Name." When you rename an object, tab, or field, the new name appears on all pages the user sees, in Salesforce for Outlook, and in Connect Offline.

Before renaming tabs, objects, fields, and other related labels, review the implementation tips for administrators.

If you use person accounts in your Salesforce organization, see Renaming Person Account Labels.

1. From Setup, enter *Rename Tabs and Labels* in the Quick Find box, then select **Rename Tabs and Labels**.
2. Select your default language from the Select Language drop-down list at the top of the page.

   Note: In Hebrew, we recommend keeping tab renaming to a minimum because variable gender in verbs is not supported and verbs can lose gender agreement.
3. Click **Edit** next to the tab you want to rename. Click **Reset** to revert to a tab's original name.

   Note: You can't reset custom object tab names.
4. Enter the singular and plural forms of the new tab name. Also, if applicable for the language, select Starts with a vowel sound for labels that start with a vowel to ensure that Salesforce uses the proper article (such as "a" or "an"). Then click **Next**.

   When you rename a tab or an object, you can't use the name of another standard tab, custom object, external object, or custom tab.
5. Enter your labels for the standard field labels and other user interface elements. Be sure to enter both a singular and plural form for each label that requires it. Select Starts with a vowel sound for labels that start with a vowel.

   Note: Some standard fields, such as Created By and Last Modified By, are purposely omitted from renaming because they track system information.
6. Click **Save**.

Repeat this procedure to translate labels into all other languages used in your organization.

Tip: After renaming a tab or object, rename any custom reports, dashboards, profiles, permission sets, custom fields, and list views that contain the original name. You can modify labels using the Translation Workbench. To rename a standard report, click **Save As** and save it to a folder designed for your new name.

Other tab customization options include:

- Individual users can control which tabs and related lists display for their own logins.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To rename a tab and field:

- "Customize Application"

  OR

  "View Setup and Configuration"

  AND

  Be designated as a translator

To reset renamed tabs:

- "Customize Application"

  OR

  "View Setup and Configuration"

  AND

  Be designated as a translator

EDITIONS

Available in: Salesforce Classic

Person accounts available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- In addition to the standard tabs provided by Salesforce, users can create entirely new custom tabs depending on their Edition. For more information, see Custom Tabs.
- In Enterprise, Unlimited, Performance, and Developer Edition organizations, you can override the tab home page that is displayed when a user clicks a standard, custom, or external object tab. For more information, see Override Standard Buttons and Tab Home Pages on page 915.

Renaming Person Account Labels

If your organization uses person accounts, you can rename these standard fields.

Field	Tab	Description
`Business Account`	Accounts	An account that is not a person account because it does not have a record type specific to person accounts. This label is primarily used to clarify the type of accounts you are importing.
`Person Account`	Accounts	A person account.
`Business Contact`	Contacts	A contact that is associated with a business account. This label is primarily used to clarify the type of accounts you are importing.

When you rename the `Person Account` field label, the renamed label appears in Salesforce:

- As a prefix to differentiate person account fields such as `Birthdate` and `Home Phone` from business account fields. For example, `Person Account: Birthdate` is available as an account column in opportunity reports.
- In the name of the `Is Person Account` field and icon. For example, if you rename the `Person Account` field to "Consumer," then `Is Person Account` becomes `Is Consumer`.

Note: The `Person Account` and `Business Account` field labels are independent from actual record type names.

- To customize person account record types, from the object management settings for person accounts, go to Record Types.
- To customize business account record types, from the object management settings for accounts, go to Record Types.

SEE ALSO:

Considerations for Renaming Tab and Field Labels

Find Object Management Settings

Considerations for Renaming Tab and Field Labels

Before renaming standard and custom tabs and fields, learn how those changes affect your users.

- Most standard tabs and objects can be renamed but not all. For example, the Forecasts tab is not available for renaming. From Setup, enter *Rename Tabs and Labels* in the Quick Find box, then select **Rename Tabs and Labels** to view a list of the tabs and objects you can rename.
- The renamed labels appear on all user pages in Salesforce including Personal Setup. However, all pages in the Setup area use the default, original labels.
- Some standard fields, such as Created By and Last Modified By, are purposely omitted from renaming because they track system information.
- After renaming tabs, objects, or fields, check the following additional items that may need manual updates:
  - Review all list view names. List view names continue to display the original object name until you change them manually.
  - Check standard report names and descriptions for the objects you renamed.
  - Update the titles and descriptions of any email templates that contain the original object or field name.
  - Manually change any other items you customized with the new object or field name. For example, custom fields, page layouts, and record types may contain the original tab or field name.
- Connect Offline, Lightning for Outlook, and Salesforce for Outlook use your new names.
- If you have renamed tabs, objects, or fields, you can also replace the Salesforce Help with another URL. Users can view this URL whenever they click on any context-sensitive help link on an end-user page or within their personal settings. After you replace the help, the **Help & Training** link at the very top of every page and all Setup pages will continue to display the Salesforce Help. For instructions on replacing the online help, see Replace Built-in Salesforce Help with Custom Help on page 56.
- In Hebrew, we recommend keeping tab renaming to a minimum because variable gender in verbs is not supported and verbs can lose gender agreement.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To rename tab and field labels:
- "Customize Application"

To reset renamed tabs:
- "Customize Application"

SEE ALSO:

Rename Object, Tab, and Field Labels

Customizing Your Pages

Page Layouts and Field-Level Security

Use field-level security to control user access to fields. Use page layouts to control the layout and organization of detail and edit pages in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal. Customize search layouts to change which fields display in search results and the buttons that display on list views.

Important: When you use page layouts to hide fields from detail and edit pages, users can still access them. Users see fields via reports, search results, list views, and the API. To restrict field access, use field-level security. Field-level security doesn't prevent searching on the values in a field. When search terms match on field values protected by field-level security, the associated records are returned in the search results without the protected fields and their values. Also don't use page layouts to secure data. For example, removing the **Edit** button from a page layout doesn't prevent users from using inline editing. To prevent users from editing data, use sharing rules, field-level security, page layout field properties, validation rules, object permissions, and Visualforce pages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts and search layouts available in: **All** Editions

Field-level security available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Field-Level Security

- Restrict users' access to view and edit fields. For example, restrict access in reports, search results, list views, related lists, email and mail merge templates, custom links, Connect Offline. Also restrict API access and when synchronizing data or importing personal data.
- Override less-restrictive field access settings in page layouts and mini page layouts. For example, if a page layout requires a field that's read-only in field-level security settings, the field remains read-only for the user.
- Override less-restrictive field settings in search layouts. For example, if a field is visible in the search layout but hidden via field-level security settings, the field remains hidden.

Page Layouts

- Control the layout and organization of detail and edit pages.
- Control which fields, related lists, and custom links users see, on detail and edit pages only.
- Control which standard and custom buttons display on detail pages and related lists.
- Determine whether fields are visible, read only, or required, on detail and edit pages only.
- Determine the fields that users can import data into.
- In Personal, Contact Manager, Group, and Professional Editions, control which fields users can access in related lists, list views, reports, Connect Offline, email and mail merge templates, custom links, and when synchronizing data.
- In Professional, Enterprise, Unlimited, Performance, and Developer Editions, determine aspects of mini page layouts, including:
  - record type
  - profile associations
  - related lists
  - fields and field access settings.

  The visible fields and related lists of the mini page layout can be further customized. However, other items inherited from the associated page layout cannot be changed on the mini page layout. Mini page layouts display selected fields and related lists of records in the mini view of the console.

Tip: To automatically add a field to all page layouts and make it visible and required everywhere regardless of field-level security, make it a universally required field.

SEE ALSO:

Page Layouts

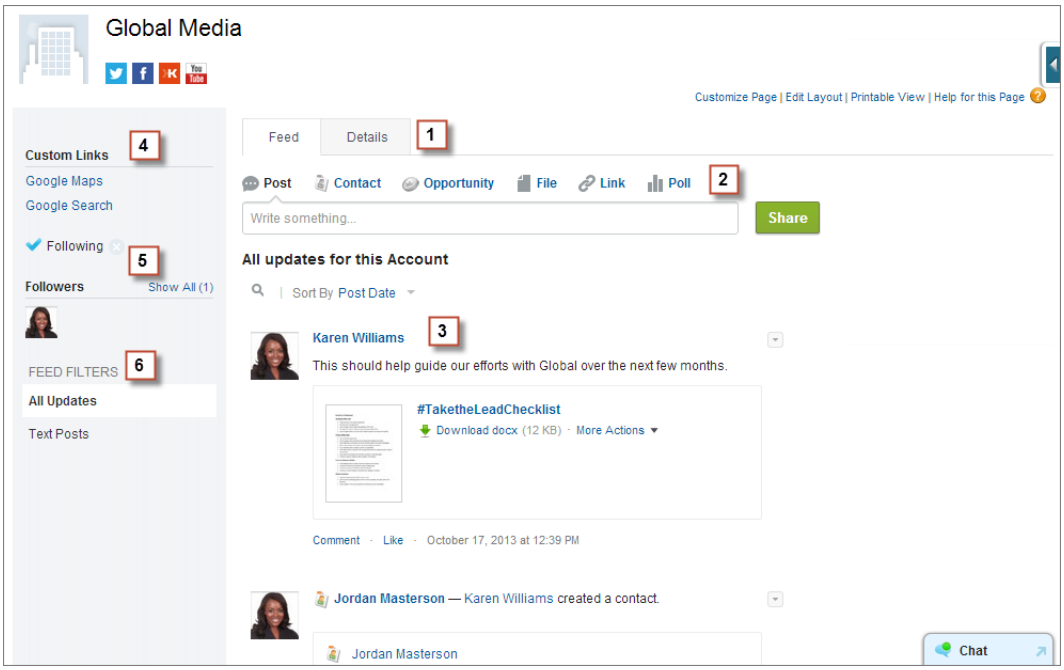
Feed-based Layouts Overview

Feed-based page layouts make it easier to work with records by providing two separate views: one for the record's feed, and one for its details, including related lists.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Unlike standard page layouts, which include all of a record's information—the feed, details, and related lists—on one page, feed-based layouts let you switch between the feed view and the details view so you can focus on the type of information you need at any given moment. For example, to see comments others have made about a record or to create a new record that's related to it, you'd use the feed view. To delve into the record's related lists, attachments, and other in-depth information, you'd use the details view.

Feed-based layouts are available on account, asset, case, contact, lead, opportunity, custom, and external objects.

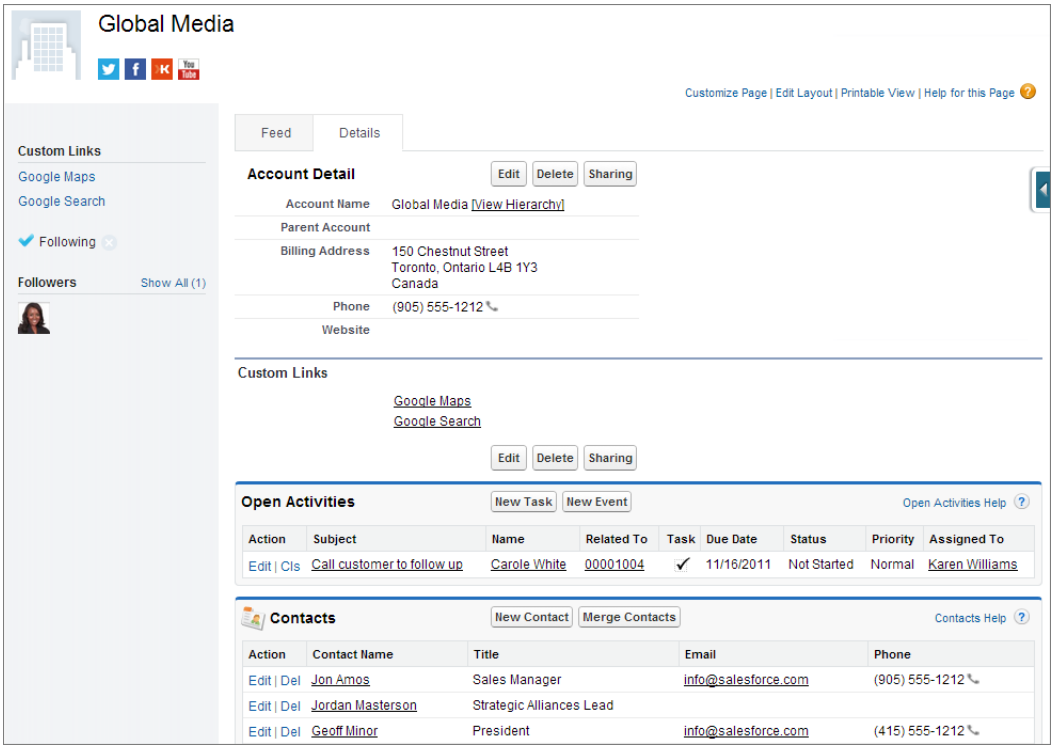


The feed view in these layouts includes:

1. Tabs or, if you're working in the Salesforce console, toggle buttons () to switch between the feed view and the detail view.
2. The publisher, which might include actions that let you do things like create related records or log calls, depending on how your administrator has set up your organization.
3. The record feed, which shows activity on the record, such as comments others have made about it.

4. Any custom buttons and links your administrator has added.
5. A follow button ( ) or following indicator ( ) and a list of people who follow the record. Depending on how your administrator has set up the page, these might appear on the left side or on the right side.
6. Feed filters, which let you choose which information from the feed you see. Depending on how your administrator has set up the page, the filters might appear on the left side of the page, in the center, or on the right.

Detail views show in-depth information about the record, including related lists.



SEE ALSO:

Create Feed-based Page Layouts

Create Page Layouts

With the enhanced page layout editor, you can tailor record page layouts to the needs of your users. Add, remove, or reorder actions, buttons, fields, and sections on a record's detail page.

1. From the management settings for the object that you want to edit, go to Page Layouts.
2. Click **New**.
3. Optionally, choose an existing page layout to clone.
4. Type a name for the new layout.
5. Optionally, select `Feed-Based Layout` to create a layout that includes separate tabs for a record's feed and detail pages.
6. Click **Save**.
7. Modify the layout.
8. Assign the new layout to user profiles.

Tip: You can also create a new page layout by cloning an existing one using Save As from within the layout.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create page layouts:
- "Customize Application"

SEE ALSO:

- Edit Page Layouts for Standard Objects
- Edit Page Layouts for Custom and External Objects
- Page Layouts and Field-Level Security
- Feed-based Layouts Overview
- Find Object Management Settings

Create Feed-based Page Layouts

Make it easier for your users to work with account, contact, lead, opportunity, custom, and external object records by creating feed-based layouts. These layouts include two separate views: one for the record's feed and one for its details.

Feed-based layouts offer a more streamlined way of working with records, and don't require users to scroll through information they're not interested in to find what they're looking for. Users can easily switch back and forth between the feed view, which includes the publisher and important events on the record, shown in chronological order, and the details view, which shows in-depth information about the record, including related lists.

You can create feed-based layouts for account, asset, contact, lead, opportunity, custom, and external objects. To create feed-based layouts for cases, use Case Feed.

1. Be sure feed tracking is enabled for the object on which you want to create a feed-based layout. To enable feed tracking, from Setup, enter *`Feed Tracking`* in the `Quick Find` box, then select **Feed Tracking**.
2. Create a new page layout and select `Feed-Based Layout`.

   Note: Only new page layouts can be feed-based; you can't change an existing standard layout to a feed-based layout.

3. Click **Edit** next to your layout and use the enhanced page layout editor to configure it.

EDITIONS

Available in: Salesforce Classic

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, and **Developer** Editions

USER PERMISSIONS

To create, edit, and delete page layouts:
- "Customize Application"

You can't configure feed-based layouts with the original page layout editor.

4. On the main page layout editor page, customize the publisher to include the actions you want to make available to users, and add any custom buttons or links.
5. Click **Feed View** in the page layout editor header to customize what appears on the feed page.

   You can:

   - Enable full-width feed so the feed expands horizontally when users view records in Salesforce console tabs or subtabs.
   - Turn on compact feed so users see a cleaner, more streamlined feed view when working with records in Salesforce console tabs or subtabs.
   - Choose to automatically collapse the publisher when it's not in use so users can see more of the information below it on the page.
   - Add custom components, which are Visualforce pages with functionality you define.
   - Choose where on the page custom buttons and links and standard components like the Follow button and followers list appear.
   - Hide the standard sidebar.
   - Choose which feed filters are available, and where they appear.
6. Assign the page layout to user profiles.

SEE ALSO:

Feed-based Layouts Overview

Page Layouts

Page layouts control the layout and organization of buttons, fields, s-controls, Visualforce, custom links, and related lists on object record pages. They also help determine which fields are visible, read only, and required. Use page layouts to customize the content of record pages for your users.

Page layouts can include s-controls and Visualforce pages that are rendered within a field section when the page displays. You can control the size of the s-controls and Visualforce pages, and determine whether or not a label and scroll bars display.

Salesforce has two drag-and-drop tools for editing page layouts: the original page layout editor and an enhanced page layout editor. The enhanced page layout editor is enabled by default, and provides all the functionality of the original editor, as well as additional functionality and an easier-to-use WYSIWYG interface.

You can enable the original page layout editor in the User Interface settings. Your Salesforce org can use only one page layout editor at a time.

From within a page layout, you can access a mini page layout. The mini page layout defines the hover details that display when you mouse over a field on an object's detail page in the Agent console or in the Recent Items section of the sidebar in Salesforce Classic.

For Personal, Contact Manager, Group, and Professional Edition orgs, every user views the same layout. Professional, Enterprise, Unlimited, Performance, and Developer Edition orgs can create different page layouts for use by different profiles and record types and set field-level security settings to further restrict users' access to specific fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Creation and deletion of page layouts is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, and delete page layouts:

- "Customize Application"

In Professional, Enterprise, Performance, Unlimited, and Developer Editions, you can set the mini page layouts and related objects that appear in the Console tab.

SEE ALSO:

- Page Layouts and Field-Level Security
- Edit Page Layouts for Standard Objects
- Edit Page Layouts for Custom and External Objects

Edit Page Layouts for Standard Objects

Change the look and feel of page layouts for standard Salesforce objects.

From the management settings for the appropriate object, go to Page Layouts.

Alternately, if you're using the Enhanced Page Layout Editor, which is enabled by default, you can customize a standard object's page layout by clicking **Edit Layout** on the object's detail page.

SEE ALSO:

- Page Layouts
- Page Layouts and Field-Level Security
- Customize Page Layouts with the Original Page Layout Editor
- Customize Page Layouts with the Enhanced Page Layout Editor

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

Edit Page Layouts for Custom and External Objects

Change the look and feel of page layouts for custom and external objects.

1. From the management settings for the object whose page layout you want to edit, go to Page Layouts.
2. Complete one of the following.
   a. If you have "Customize Application" permission, open the page layout that you want to customize for edit.
   b. If you have the "View Setup and Configuration" permission, click the page layout that you want to view.

SEE ALSO:

- Page Layouts
- Page Layouts and Field-Level Security
- Customize Page Layouts with the Original Page Layout Editor
- Customize Page Layouts with the Enhanced Page Layout Editor
- Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

Customize Page Layouts with the Enhanced Page Layout Editor

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

The enhanced page layout editor is a WYSIWYG tool that allows you to customize your organization's page layouts for detail and edit pages in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal. The enhanced page layout editor has all the functionality of the original page layout editor, but has more features and an easier-to-use interface.

The enhanced page layout editor has two parts: a palette on the upper portion of the screen and the page layout on the lower portion of the screen. The palette contains the user interface elements that you can add to your page layout, such as fields, actions, buttons, links, and related lists.

When working with the enhanced page layout editor:

- To select multiple elements individually, use CTRL+click. To select multiple elements as a group, use SHIFT+click.
- To change the properties of an element on the page layout, double-click the element or click the wrench icon ( ) next to it. You can't change the properties of elements in the palette.
- To make a field read-only or required, double-click the field in the page layout and select the appropriate checkbox.
- To access the other layouts for an object with multiple page layouts, click the page layout name at the top of the page and select another layout to view.
- To change the name of the page layout, add personal and public tags if available, and display standard object checkboxes on the page layout, click **Layout Properties**.

  Note: You can't rename a page layout if you're using Salesforce Professional Edition.

- In Enterprise, Unlimited, Performance, and Developer Editions, you can select a profile to preview how the pages will look for users with that profile. Most related lists' columns preview without data.
- If you're working with a feed-based page layout, click **Feed View** to customize the tools and components that appear when users are working in the feed on a record.
- To choose which fields display on the record detail page and the order in which they appear, click **Edit Multi-Line Layout**.
- The mini page layout defines the hover details that display when you mouse over a field on an object's detail page, in the Agent console, or in the Recent Items section of the sidebar in Salesforce Classic. To customize the fields in the mini page layout, click **Mini Page Layout** at the top of the palette.
- When you're done customizing the page layout, save it. If you navigate away from your page layout before saving, you lose your changes.

SEE ALSO:

Guidelines for Using the Enhanced Page Layout Editor

Customize Related Lists

Feed-based Layouts Overview

Page Layouts

Enhanced Page Layout Editor User Interface Elements

This list describes the enhanced page layout editor user interface elements and how you can use them in your page layout. To add elements, drag them from the palette to the layout. Valid drop locations show up in green. To remove elements, drag them off the layout and back to the palette.

Tip: Create the appropriate buttons, custom links, fields, custom s-controls, and Visualforce pages before editing your page layout.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

Actions

Mobile smart actions appear as a single action element in the page layout editor. However, they appear in the action bar and action menu in Salesforce1 as distinct create actions. These distinct actions allow users to create records directly from the action bar. The create actions included in the set of mobile smart actions vary depending on the page layout's object.

Note: If you delete an action, the action is removed from all layouts that it's assigned to.

Wave Analytics Assets

You can add and move a Wave Analytics dashboard to any section on the page layout, except Mobile Cards.

For a Wave Analytics dashboard element, use field mapping to map data fields in the dashboard to the object's fields so that the dashboard shows only the data that's relevant for the record being viewed. You can specify fields from multiple datasets. Identify each dataset by its system name, which is listed on the dataset's edit page. Identify the data fields by row, and for the values, specify either the field names from the Salesforce object or specific values. The JSON field mapping follows this format:

```
{
    "dataset1_system_name": {
        "row_name": ["$field_name"],
        "row_name": ["$field_name"]
    },
    "dataset2_system_name": {
        "row_name": ["$field_name"],
        "row_name": ["$field_name"]
    }
}
```

For example, if the dashboard shows data from a dataset named Service, with the dimensions Account and Industry, the field mapping would be defined as:

```
{
    "service": {
        "account": ["$Name"],
        "industry": ["$Industry"]
    }
}
```

Blank Spaces

You can add and move blank spaces to any section on the page layout, except Mobile Cards. Use blank spaces to visually align and distinguish elements on the page.

Note: If you use the original page layout editor to view a page layout that was created in the enhanced page layout editor, the original page layout editor shows the blank spaces you added. You can't move or add blank spaces in the original page layout editor, but you can remove them by dragging them to the box on the right.

Blank spaces aren't supported in Lightning Experience.

Buttons

You can control which standard and custom buttons are displayed and the order in which the custom buttons appear. You can't rearrange the order of the standard buttons.

Standard and custom buttons are available as actions in Salesforce1 and Lightning Experience.

Canvas Apps

For the Canvas Apps category to appear in the palette, set the canvas app location to Visualforce Page when you create the canvas app in Salesforce.

- If you add a canvas app to any section other than the Mobile Cards section, the canvas app appears in the page layout in the full Salesforce site or in the record detail page in Salesforce1.
- If you add a canvas app to the Mobile Cards section, the canvas app appears as a card on the record related information page in Salesforce1.

Components

When you add a component to the Mobile Cards section of a page layout, it displays as a card on a record's related information page in Salesforce1. Components are supported only in the Mobile Cards section of a page layout, and they only appear in Salesforce1.

In organizations that are created after Spring '14, the Twitter component is added by default to the Mobile Cards section of page layouts for objects that support it.

Note: Social Accounts and Contacts must be enabled for your organization before you can add the Twitter component to a page layout.

The Twitter card that you see on account, contact, and lead record pages in Lightning Experience isn't the same as the Twitter component available as a mobile card in the page layout editor. The Twitter card in Lightning Experience is a Lightning component. You must have Social Accounts and Contacts enabled for it to appear.

Expanded Lookups

The Expanded Lookups category for the enhanced page layout editor contains fields with lookup relationships. These fields, when added to the Mobile Cards section of the layout, display as related lookup cards on a record's related information page in Salesforce1. A page layout can have up to 20 expanded lookups.

The Expanded Lookups category only contains fields with lookup relationships to objects that support compact layouts.

Expanded lookups aren't supported in Lightning Experience.

Fields

A field can display one or more of these icons:

- The field must have a value to save the record, but isn't required on the page layout itself.
- The field must be included on the page layout because either an administrator configured the field as universally required or Salesforce automatically requires the field. Although you can't remove such fields, you can move them to different locations.
- The field is a controlling field.
- The field is a dependent field.
- The field is read only.

To set which fields are required and read only, select one or more fields and click the wrench icon ( ) on any of the selected fields.

- The field properties of some standard fields cannot be changed. Custom fields can only be changed if they are not universally required fields.
- Fields marked as read only are always editable by administrators and users with the "Edit Read Only Fields" permission.
- If you make a picklist field read only, all new records contain the default value for that picklist.
- Auto-number fields are always read only.

- If you mark the opportunity `Probability` field as read only, the `Probability` value is still updated when a user changes the `Stage` value of an opportunity.

When working with fields:

- In Personal, Contact Manager, and Group Editions, page layouts control which fields users can access in related lists, list views, reports, Connect Offline, email and mail merge templates, custom links, and when synchronizing data. In Professional, Enterprise, Unlimited, Performance, and Developer Editions, field-level security controls this access. Field-level security settings override field properties that you set on the page layout if the field-level security is more restrictive than the page layout setting.
- Users can import values into a field only if they have read and edit access. Field access is determined by user permissions, page layout assignments, and field-level security settings.

Related Lists

A page layout can have up to 100 related lists. You can place related lists at the bottom of the page layout. To move a related list on the page layout, drag the handle located above the related list.

To customize a related list, double-click the related list handle or click the wrench icon ( ) inside the handle. Use the related list properties to:

- Specify which fields display as columns on the related list, the order in which they appear, and the sort order of the records in the related list. In Professional, Enterprise, Unlimited, and Performance Editions, you can also opt to apply the column information to other page layouts for the same type of object.
- Specify which standard and custom buttons appear on the related list.

When working with related lists on page layouts, note the following:

- The **View All** button only displays up to 2,000 items in a related list.
- Some related lists aren't customizable because they link to data rather than store it. Salesforce denotes related lists that aren't customizable on the page layout.
- You can't add related lists to the page layouts for the User object.
- You can also enable related list hover links for your organization so that record detail pages include links for each related list at the top of the page. Users can hover the mouse over a related list hover link to display the corresponding related list in an interactive overlay that allows them to quickly view and manage the related list items. Users can also click a related list hover link to jump down to the content of the related list without scrolling down the page.
- In Professional, Enterprise, Unlimited, Performance, and Developer Edition, individual users can customize which related lists display for their personal use. Administrators can overwrite these user customizations and apply the related list configuration in the page layout to all users, even if they already customized their display. To overwrite users' related list customizations, click **Yes** on the Overwrite Users' Customized Related Lists popup window, which appears when saving a page layout if you moved or added a related list.
- Related lists show up on the record related information page in Salesforce1.

Custom S-Controls

A page layout can have up to 20 s-controls.

To change the properties of an s-control, double-click the s-control or click its wrench icon ( ) and set the following attributes:

- `Width` sets the horizontal size in pixels or a percent.
- `Height` sets the vertical size in pixels.
- `Show scrollbars` determines whether the iFrame in which the s-control displays contains scroll bars when necessary.
- `Show label` determines whether the page layout includes the `Label` of the custom s-control. Remove the label to display the s-control in a wider area.

S-controls aren't supported in Lightning Experience.

Sections

You can add and move sections anywhere above the related lists on the page layout. The sections you add can contain fields, s-controls, and blank spaces. In addition, each page layout has a default section that can only contain custom links and blank spaces. You can change the location of the custom link section, but you can't remove it from the page.

The Section user interface element is the second option in the palette when you select the Fields or Custom S-Controls category on the palette.

To change the attributes of a section, double-click the section or select its associated wrench icon ( ). You can:

- Enter a name for the section. Names of some standard page sections cannot be changed.
- Specify whether the section has one or two columns.
- Specify the order in which users can tab through the items in that section.
- Specify whether the section heading is shown on the detail and edit pages.

Tags

If tags are enabled in your organization, click **Layout Properties** and use the checkboxes to indicate whether personal and public tags are included in the header section of the page layout. Users can't tag a record if neither personal nor public tags are included in the header section. Also, the positioning of personal and public tags in the header cannot be modified.

Tags aren't supported in Lightning Experience.

Visualforce Pages

Visualforce pages can be added to any section on the page layout except for sections reserved for custom links and related lists. A page layout can have up to 20 Visualforce pages.

You can add a Visualforce page to a page layout only if the standard controller on the Visualforce page is set to the object for which you are creating the page layout. If you don't have any Visualforce pages with a standard controller set to that object, the Visualforce Pages category doesn't appear in the palette.

Only Visualforce pages with the `Available for Salesforce mobile apps and Lightning Pages` checkbox selected will display in Salesforce1.

Mobile-enabled Visualforce pages show up as slightly differently colored elements in the palette than their non-mobile-enabled counterparts. Hovering your mouse over a Visualforce page element in the palette shows whether the Visualforce page is mobile-enabled.

SEE ALSO:

Guidelines for Using the Enhanced Page Layout Editor

Customize Page Layouts with the Enhanced Page Layout Editor

Guidelines for Using the Enhanced Page Layout Editor

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

- Elements that are already on the page layout still appear on the palette but are inactive. When you click an inactive element on the palette, Salesforce highlights the element on the page layout.
- Removing a field from a page layout doesn't remove it from the object's compact layout. The two layout types are independent.
- If the original page layout editor is enabled, users can click the page layout name to access the detail page of the page layout. The enhanced page layout editor doesn't have detail pages, as all the detail page functionality is always available on the enhanced editor. Salesforce displays a read-only version of the enhanced page layout editor to users with the "View Setup and Configuration" permission.

  Note: The read-only view of the page layout doesn't display field types and lengths in hover details.

- The Custom Links, Custom S-Controls, and Visualforce Pages categories only appear in the palette if you have defined those types of elements for the object for which you are defining a page layout. When you create a custom link for an object, you add it to the Custom Links section on that object's page layout. In non-English Salesforce organizations, the "Custom Links" section title is not translated from English automatically for the Territory and Territory Model objects, but you can edit the section title.
- The Canvas Apps category appears in the palette only if you defined at least one canvas app with a location of Visualforce Page.
- The Components category appears in the palette only if the available components are supported by the object for which you are defining a page layout. For example, the Twitter component is supported only on account, contact, and lead page layouts.
- When editing a person account page layout:
  - If you add `Shipping Address` next to `Billing Address` in the Address Information section, a link displays on the person account edit page that allows you to copy the billing address to the shipping address. Also, an equivalent link appears if you add `Other Address` to the Address Information section.
  - Contact fields and related lists are available on person account page layouts, but contact custom links and custom buttons are not.
- This table lists standard objects that have checkboxes that are specific to page layouts for that object. To configure how Salesforce displays the checkboxes, click **Layout Properties** when customizing the page layout. Use the `Select by default` checkbox associated with a checkbox if you want Salesforce to automatically select the option when a user accesses the edit page.

Object	Checkboxes
Account	`Evaluate this account against territory rules on save` checkbox — Displays the `Evaluate this account against territory rules on save` checkbox on account edit pages. Territory assignment rules run automatically when the `Select by default` checkbox is selected. If both `Show on edit page` and `Select by default` are selected, users can uncheck the `Evaluate this account against territory rules on save` checkbox on the account edit page, and territory assignment rules will not be run.

Case	– `Case assignment checkbox` — Displays the `Assign using active assignment rules` checkbox on case edit pages. Case assignment rules run automatically when the `Select by default` checkbox is selected. If both `Show on edit page` and `Select by default` are selected, the assignment checkbox is selected by default, but users can deselect it to override the assignment rule. – `Email notification checkbox` — Displays the `Send notification email to contact` checkbox on case edit pages.
Case Close	– `Solution information section` — Displays the solution information section on the case close edit pages. – `Notify Contact` — Displays the `Notify Contact` checkbox on case close edit pages.
Lead	`Lead assignment checkbox` — Displays the `Assign using active assignment rule` checkbox appears on the lead edit page. Lead assignment rules run automatically when the `Select by default` checkbox is selected. If both `Show on edit page` and `Select by default` are selected, the assignment checkbox is selected by default, but users can deselect it to override the assignment rule.
Person Account	`Evaluate this account against territory rules on save` checkbox — Displays the `Evaluate this account against territory rules on save` checkbox on person account edit pages. Territory assignment rules run automatically when the `Select by default` checkbox is selected. If both `Show on edit page` and `Select by default` are selected, users can uncheck the `Evaluate this account against territory rules on save` checkbox on the account edit page, and territory assignment rules will not be run.
Task	`Email notification checkbox` — Displays the `Send Notification Email` checkbox appears on the task edit page. Note: A user's personal preference for defaulting the state of the checkbox takes precedence over the organization-wide setting.

SEE ALSO:

Enhanced Page Layout Editor User Interface Elements

Customize Page Layouts with the Enhanced Page Layout Editor

Page Layout Tips

Here are a few tips to keep your page layouts organized and easy to use.

- Use field-level security to restrict users' access to fields; then use page layouts to organize detail and edit pages within tabs. This reduces the number of page layouts for you to maintain. Field-level security settings override the visible and read-only settings on the page layout if the field-level security has a more restrictive setting than the page layout.
- Remove unnecessary fields.
- Keep the number of required fields to a minimum.
- Group similar fields with sections.
- Think about the right TAB key order for each section.
- Check your layouts in Read and Edit modes.
- Add help and description text to custom fields. Use it to explain to users what data you're looking for in the field.
- In Professional, Enterprise, Performance, Unlimited, and Developer Editions, use record types to provide unique layouts for different records.
- Optimize related lists—adjust their overall order, the sorting of the records, and display of relevant columns and buttons.
- If you want to customize the user profile layout in Salesforce1, create a new layout or edit an existing layout in the User Profile Page Layouts section.
- If a dependent lookup is above its controlling field on a page layout, make its lookup filter optional or redesign the page layout. Placing a required dependent lookup above its controlling field on a page layout could confuse users who typically start from the top of the page when entering data.
- A background process periodically runs that cleans up metadata associated with deleted custom fields. This process will affect the `Last Modified Date` and `Last Modified By` fields on page layouts, record types, and custom objects.
- Salesforce recommends creating no more than 200 page layouts. Although there is no limit, it can be difficult to manage your page layouts if you have more than 200.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Creation and deletion of page layouts is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Page Layout Considerations

Keep these considerations in mind when working with page layouts in the enhanced page layout editor.

Page Layouts

- You can't rename a page layout if you're using Salesforce Professional Edition.
- You can drag up to 20 s-controls, 20 Visualforce pages, 20 expanded lookups, and 100 related lists onto a page layout. There are no limits on fields and custom links. You can add one Wave Analytics dashboard per page layout.

  Note: You can't place a Visualforce page more than once on a page layout.

- Page layouts for the user object only include custom fields, custom links, s-controls, and Visualforce pages. Tagging, related lists, custom buttons, and standard field customizations are *not* included on page layouts for the user object. Also, field-level security is only available for custom fields on the user object. Only standard Chatter actions (Post, File, Link, Poll, and Thanks) appear on the user profile page, regardless of the actions in the User page layout.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Creation and deletion of page layouts is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- You can edit only certain attributes when you're working with a page layout that was installed from a managed app. Some changes that you make to managed page layouts, such as adding components, work when you're editing the page layout but aren't reflected on the record detail page.
- Chatter group layout changes affect the Salesforce1 app only. Changes to the group publisher (actions and layout) reflect in both the full Salesforce site and the Salesforce1 app.

Mini Page Layouts

- You can't choose **Mini Console View** for the Close Case layout, Log a Case page, or View Cases page layouts on the Self-Service Portal. You can't choose **Mini Console View** for opportunity team page layouts.
- Field properties on the page layout determine field properties on the mini page layout. For example, if a field is read-only on the page layout, that same field will be read-only on the mini page layout. To change the field properties of fields on the mini page layout, you must change the field properties of fields on the page layout.
- You can't define mini page layouts for the Close Case layout, Log a Case page, or View Cases page layouts on the Self-Service Portal. You can't define mini page layouts for opportunity team page layouts.
- You can define mini page layouts for the user object; however, you cannot add standard fields or related lists. Also, a customized mini page layout won't display in the Agent console.
- Overrides for the **Edit** and **View** buttons for an object don't affect the **Edit** and **View** buttons in mini page layouts.
- Fields marked `Always Displayed` or `Always on Layout` on page layouts are automatically included on the mini page layout and cannot be removed unless they are removed from the page layout.

Customizing Home Tab Page Layouts

You can customize the Home tab to include components such as sidebar links, a company logo, a dashboard snapshot, or custom components that you create. A dashboard snapshot is a clipping of the top row of a dashboard's components. Just like other tabs, you can also assign different home page layouts to different users based on profile.

You can add components to the sidebar or the main panel. You can also determine if custom sidebar components appear only on the Home tab or on all Salesforce pages.

To start customizing your home page layouts, see:

- Create Custom Home Page Components on page 23
- Design Home Page Layouts on page 24
- Assign Home Tab Page Layouts to Profiles on page 26

EDITIONS

Available in: Salesforce Classic

Available in all editions

USER PERMISSIONS

To view home page layouts:
- "Customize Application"

To create or change home page layouts:
- "Customize Application"

Create Custom Home Page Components

Use custom components to configure the Salesforce Classic home page for your users. Add HTML, images, links, and more to enhance your users' productivity.

EDITIONS

Available in: Salesforce Classic

Available in all editions

USER PERMISSIONS

To create or change home page layouts:
- "Customize Application"

Before you begin:

- If you're creating custom link components, define your Home tab custom links first. See Custom Buttons and Links on page 904.
- If you're creating an image component, upload your image to the Documents tab first.
- If you're creating a Visualforce Area component, create your Visualforce page first.

1. From Setup, enter *Home Page Components* in the Quick Find box, then select **Home Page Components**.
2. Click **New**.
3. Enter a name for the component. For custom links, this name is displayed as the section heading in the sidebar on the Home tab.
4. Choose the type of component.
5. Click **Next**, and then complete one or more of these steps.
   - For links, select the appropriate custom links, and then click **Add**.
   - For images, click **Insert an image**, choose the document folder, and then select the image file. The image file must be in a public folder and Externally Available must be enabled on the document's properties so that users can view the image.

     Tip: Keep your image size smaller than 20 KB for optimum performance.

   - For an HTML Area component, choose where to display it—in the wide or narrow column—and then enter your content in the box below.

     Note: HTML Area home page components don't support JavaScript, CSS, iframes, and some other advanced markup. To use JavaScript or other advanced HTML elements in your home page component, we recommend that you use a Visualforce Area component instead.

   - For a Visualforce Area component, choose where to display it—in the wide or narrow column—then select the Visualforce page, and assign it a height.
6. Click **Save**.

After creating the home page component, you need to add it to a home page layout. See Design Home Page Layouts on page 24.

Note: Components in the narrow column are displayed in the sidebar. They aren't displayed in the sidebar on other pages in Salesforce unless you specify that in your user interface settings or by assigning the "Show Custom Sidebar On All Pages" permission.

SEE ALSO:

Visualforce Area Home Page Components

Home Page Components Tips and Considerations

Visualforce Area Home Page Components

Use Visualforce Area home page components to add dynamic content to your home page. For example, you can present content from partner apps, display charts with the Reports and Dashboards REST API, or add a canvas app to the home page.

The Visualforce page that you choose for the component can use a standard or custom controller. You can include JavaScript in your Visualforce page, but because the component is rendered in an iframe on the home page layout, the JavaScript can't interact with the page that contains the component.

EDITIONS

Available in: Salesforce Classic

Available in all editions

Sample Usage

If your Visualforce Area home page component displays in the sidebar, you can dynamically get the record ID and top-level URL of the page that the component is being displayed on by using the `$CurrentPage` global variable in your Visualforce markup.

Using `$CurrentPage`, you can access the query string parameters for the page by specifying the `parameters` attribute, after which you can access each individual parameter:

```
$CurrentPage.parameters.parameter_name
```

The parameters for record ID and top-level page URL are, respectively, `id` and `sfdcIFrameOrigin`. For more information, see "Getting Query String Parameters" in the *Visualforce Developer's Guide*.

Design Home Page Layouts

After creating the components you want displayed on the Home tab, design your home page layouts. You can design your layouts based on your unique organizational and user needs.

1. From Setup, enter *Home Page Layouts* in the `Quick Find` box, then select **Home Page Layouts**.
2. Click to edit an existing layout or create one. Alternately, select a layout to copy and click **Clone**.
3. If you create a layout, give it a name and then click **Save**.
4. Select the components to display on the layout.
   - To add the Find Articles component, select **Article Search**. This component is only available for Salesforce Knowledge users.
   - To add the Customer Portal component, select **Customer Portal Welcome**. If the My Profile site Visualforce page has been enabled, this component contains a personalized welcome message and a link to the portal user's profile. The My Profile page enables users logged into either your Force.com site, or your Customer Portal from Force.com sites, to update their own contact information. When they make changes to this page, the corresponding portal user and contact records are updated.
   - To allow your users to resume flow interviews that they've paused, select **Paused Flow Interviews**. This component displays only flow interviews that the user has paused.
5. Click **Next**.
6. Customize the order in which the narrow and wide components appear. Move a component by selecting it and using the arrow buttons.

EDITIONS

Available in: Salesforce Classic

Available in all editions

USER PERMISSIONS

To view home page layouts:
- "View Setup and Configuration"

To create or change home page layouts:
- "Customize Application"

7. Click **Save**.

SEE ALSO:

Assign Home Tab Page Layouts to Profiles

Customizing Home Tab Page Layouts

Home Page Components Tips and Considerations

Keep these considerations in mind when creating custom components that you want displayed on the Salesforce Classic Home tab.

EDITIONS

Available in: Salesforce Classic

Available in all editions

- Standard components without an **Edit** link are read only.
- The components that you select for the narrow column display in the sidebar. They don't display in the sidebar on other pages within Salesforce unless you specify that in your user interface settings. If you only want certain users to view sidebar components on all pages, grant those users the "Show Custom Sidebar On All Pages" permission.
- When editing the standard Messages & Alerts component, enter the text that you want to display to users. If entering HTML code for your message, make sure that it's self-contained, well-formed HTML.

  Note: Standard Messages & Alerts home page components don't support JavaScript, CSS, iframes, and some other advanced markup.

- When editing the standard Custom Links home page component, enter the link text to display to users in the `Bookmark` field. In the `URL` field, enter the complete website address, such as `http://www.yahoo.com`. To link to a Salesforce page, enter only the part of the URL after `salesforce.com`, for example, `/000x0000000esq4`. These links always open within the main Salesforce window, not in a popup window.
- The standard Custom Links home page component is a quick way to add links to the sidebar, but it doesn't support merge fields, functions (such as URLFOR), executing JavaScript, or customizable window opening properties. If you need this additional functionality:
  1. From Setup, enter *Home* in the `Quick Find` box, then select **Custom Links**, and then create your home page custom links on that page.
  2. From Setup, enter *Home* in the `Quick Find` box, then select **Home Page Components**, and then create a custom home page component of type Links on that page that includes the custom links that you created in the first step. Creating a custom home page component for your links doesn't change the visual styling for your end users.
- The `Dashboard Snapshot` component displays the top three components of the last dashboard the user accessed. Users can view a dashboard snapshot on their Home tab if they have access to at least one dashboard.
- When designing home page layouts for your Customer Portal, we recommend adding the following components: Search, Solution Search, Recent Items, Customer Portal Welcome, and a custom HTML Area component that includes your corporate branding in the wide column.

SEE ALSO:

Create Custom Home Page Components

Assign Home Tab Page Layouts to Profiles

Your home page layouts are only visible to users after you assign them to a user profile.

1. From Setup, enter *Home Page Layouts* in the Quick Find box, then select **Home Page Layouts**.
2. Click **Page Layout Assignment**.
3. Click **Edit Assignment**.
4. Choose the appropriate page layout for each profile.

   Initially, all users, including Customer Portal users, are assigned to the Home Page Default layout.
5. Click **Save**.

Tip: Users can customize the dashboard settings on their Home tab in their personal settings.

EDITIONS

Available in: Salesforce Classic

Available in all editions

USER PERMISSIONS

To assign home page layouts:

- "Customize Application"

Set Page Layouts and Field-Level Security

Use field-level security as the means to restrict users' access to fields; then use page layouts primarily to organize detail and edit pages within tabs. This reduces the number of page layouts for you to maintain. For example, if a field is required in the page layout and read only in the field-level security settings, the field-level security overrides the page layout and the field will be read only for the user.

Note: Field-level security doesn't prevent searching on the values in a field. When search terms match on field values protected by field-level security, the associated records are returned in the search results without the protected fields and their values.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts and search layouts available in: **All** Editions

Field-level security available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

For Personal, Contact Manager, and Group Editions

1. Create custom fields.
2. Create any custom buttons or links.
3. Create any custom s-controls.
4. Define page layouts. All users automatically use the same page layout for each object.
5. Set the related objects and the mini page layouts that display in the console.
6. Define search layouts. All users use the same search layouts.

For Professional, Enterprise, Unlimited, Performance, and Developer Editions

1. Create custom fields.
2. Create any custom buttons or links.
3. Create any custom s-controls.
4. Create any custom profiles.
5. Create record types for different business scenarios.
6. Assign which record types are available to users with different profiles.
7. Set the field-level security for each profile to restrict users' access to specific fields.
8. Define page layouts to organize your pages.
9. Set the related objects and the mini page layouts that display in the console.

10. Assign page layouts to users based on profiles and record types.
11. Check the field accessibility grid to verify that all field access settings are correct.
12. Define search layouts. All users use the same search layouts.

Tip: Click **Preview** while editing a page layout to see how the page will look for users with different profiles. This preview includes any extra security that is set in field-level security.

Assigning Page Layouts

After defining page layouts, assign which page layouts users see. A user's profile determines which page layout he or she sees. In addition, if your organization is using record types for a particular tab, the combination of the user's profile and the record type determine which page layout is displayed.

You can assign page layouts from:

- The object's customize page layout or record type page
- The original or enhanced profile user interface.

To verify that users have the correct access to fields based on the page layout and field-level security, you can check the field accessibility grid.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Record types are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Page Layouts

Record Types

Page Layouts and Field-Level Security

Assign Page Layouts from a Customize Page Layout or Record Type Page

1. From the management settings for the appropriate object, go to Page Layouts or Record Types.
2. Click **Page Layout Assignment**.
3. Click **Edit Assignment**.
4. Use the table to specify the page layout for each profile. The table displays the page layout assignments for each profile. If your organization uses record types, a matrix displays a page layout selector for each profile and record type.

   When selecting page layout assignments:

   - Click a cell, column, or row heading to select all the table cells in that column or row.
   - Press SHIFT+click to select multiple adjacent table cells, columns, or rows.
   - Press CTRL+click to select multiple nonadjacent table cells, columns, or rows.
   - Click any cell and drag to select a range of cells.
   - Click **Next** or **Prev** to view another set of record types.

   Selected page layout assignments are highlighted. Page layout assignments you change are italicized until you save your changes.
5. If necessary, select another page layout to assign from the `Page Layout To Use` drop-down list and repeat the previous step for the new page layout.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Record types are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign page layouts:

- "Manage Profiles and Permission Sets"

6. Click **Save**.

SEE ALSO:

Assigning Page Layouts

Edit Multi-Line Layouts for Opportunity Products

You can customize the columns that display when you click **Edit All** in the Products related list of an opportunity detail page.

1. From the object management settings for opportunity products, go to Page Layouts.
2. Next to the name of an opportunity product page layout, click **Edit**.
3. Click **Edit Multi-Line Layout**.
4. Move fields between Available Fields and Selected Fields.
   - To customize which fields display in the layout, select one or more fields in Available Fields and click **Add** or **Remove**.
   - To sort fields in the layout, select one or more fields in Selected Fields and click **Up** or **Down**.
   - To select multiple fields individually, use CTRL+click.
   - To select multiple fields as a group, use SHIFT+click.
5. Click **Save** to apply your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit multi-line layouts for opportunity products:
- "Customize Application"

Build Page Layouts for Custom Objects

Page layouts define which fields users can view and edit when entering data for a custom object record. You can use the default page layout that is created automatically when you create a custom object. You can also build your own page layout with related lists and custom links. If you do not use any page layout with your custom object, you can still interact with it by using the Force.com API to manage custom data or build a custom user interface.

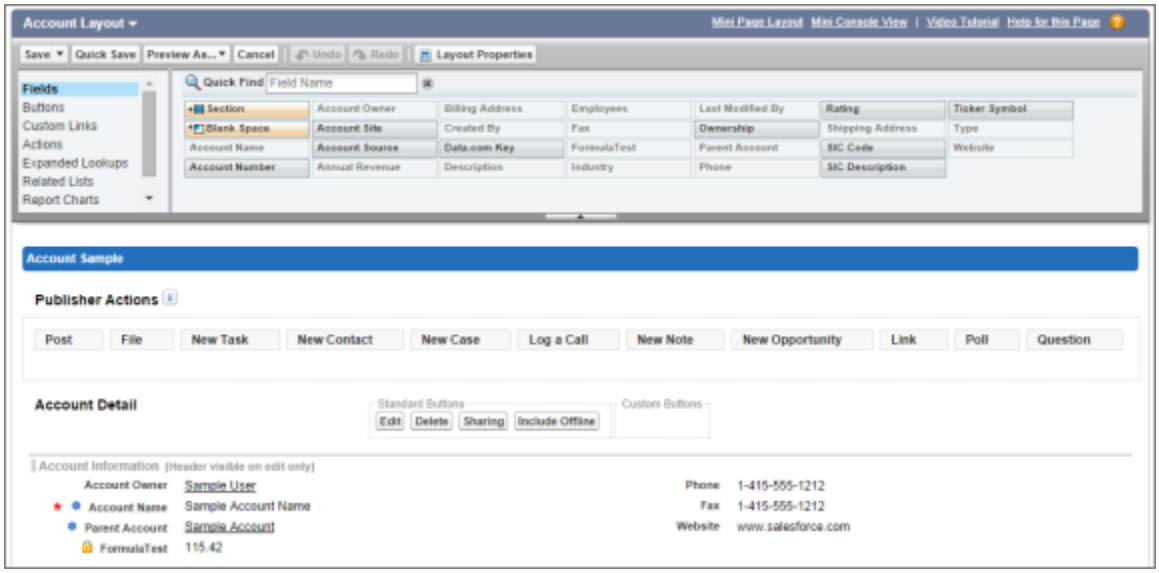
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To define page layouts for custom objects:
- "Customize Application"



1. From the object management settings for a custom object, go to Page Layouts.
2. In the Page Layouts related list, open an existing page layout for edit, or create a new page layout.
3. If creating a new page layout, enter a name for it, and select an existing page layout to clone, if desired.
4. Edit the page layout just as you would any other page layout.
5. Click **Save**.

SEE ALSO:

Page Layouts

Find Object Management Settings

How Page Layouts Work in Salesforce1

Use the enhanced page layout editor to customize the layout of an object's record detail pages, configure actions, and adjust which fields and related lists appear in Salesforce1.

In Salesforce1, page layouts drive these areas of the mobile experience.

Record Related Information and Detail Pages

When you view a record in Salesforce1, you see the fields, Visualforce pages, and related lists that are based on the record type and the user's profile. Related lists show up as single-line cards containing the name of the page or related list. Tapping the related list card displays its details.

Mobile Cards

You can add expanded lookups, components, canvas apps, and Visualforce pages to the Mobile Cards section of your page layout to have them show up as mobile cards in Salesforce1. The elements you place in this section don't show up on a record's detail page in the full Salesforce site. They appear only on the record's related information page in Salesforce1.

Note: In organizations that are created after Spring '14, the Twitter component is added by default to the Mobile Cards section of page layouts for objects that support it.

EDITIONS

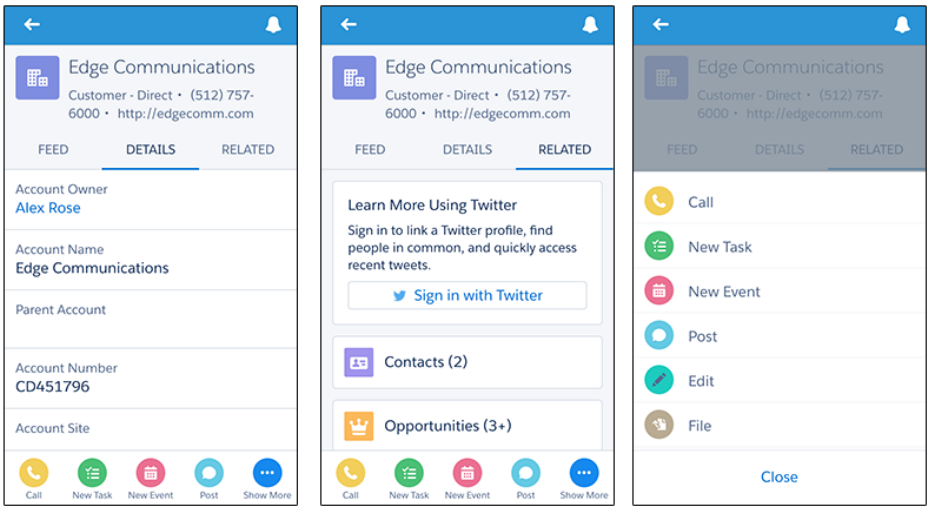
Available in: both Salesforce Classic and Lightning Experience

Available in: **All** editions except Database.com

Actions

In Salesforce1, actions in the Salesforce1 and Lightning Experience Actions section of a page layout appear in the action bar and action menu on the object's record pages.

Here are the record detail page, related information page, and action menu for a sample account, Edge Communications:



Page Layouts in Lightning Experience

When you customize your object record pages in Salesforce Classic, they affect the content of object record pages in Lightning Experience. However, in Lightning Experience, the page elements display differently, and some aren't supported.

If you're in an organization that supports multiple page layouts, you can create page layouts directly from the Page Layouts related list on any object in the Object Manager. You can also edit or delete an object's page layouts by clicking ▼ on a page layout in the Page Layouts related list.

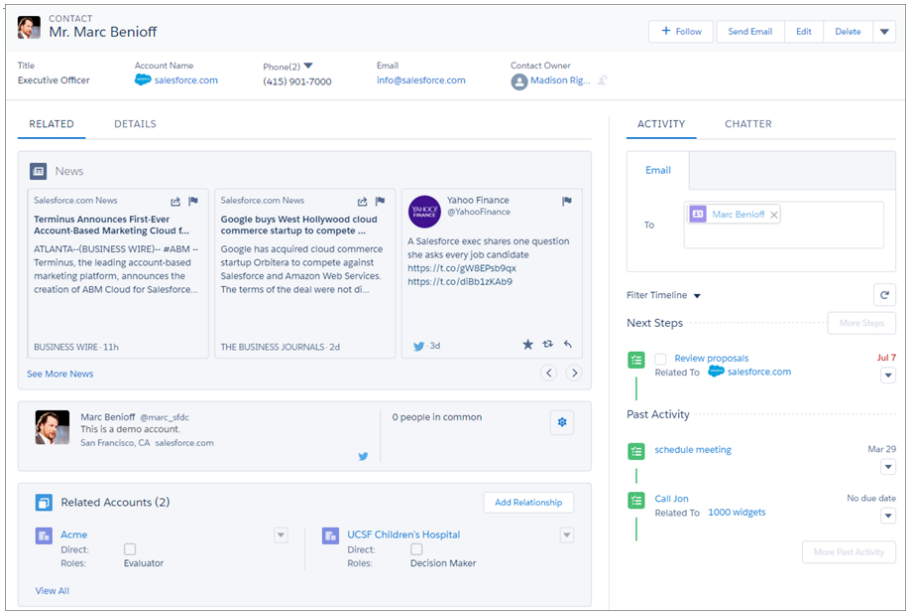
Here's a sample contact record in Lightning Experience. The highlights panel, which contains key fields for the record, is the only part of a record page that you can't customize using the page layout editor. The fields in the highlights panel are customized using a compact layout.

EDITIONS

Available in: Lightning Experience

Page layouts are available in: **All** Editions

Creation and deletion of page layouts is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



These page layout elements are supported in Lightning Experience.

Actions

Actions display in different places, such as the highlights panel, Activity tab, and the Collaborate tab. The actions are derived from the list of actions in the Salesforce1 and Lightning Experience Actions section of the page layout. Some actions aren't supported in Lightning Experience.

For more information, see Actions in Lightning Experience on page 886.

Canvas Apps

Canvas apps are supported in Lightning Experience.

Custom Links

Custom links display under the Details tab.

Fields

Fields display under the Details tab. You can remove or reorder fields on a page layout only via the page layout editor.

Related Lists

Related lists are included as Lightning components in Lightning Experience. Not all related lists are supported in Lightning Experience. For example, the `Object` History related list isn't supported.

Standard and Custom Buttons

Standard and custom buttons are treated as actions in Lightning Experience, just like in Salesforce1.

Important: Custom buttons that call JavaScript aren't supported in Lightning Experience.

Visualforce Pages

Visualforce pages that you've added to the page layout appear under the Details tab. Only Visualforce pages with `Available for Salesforce mobile apps and Lightning Pages` enabled display in Lightning Experience.

Visualforce pages that have been put into the Mobile Cards section as components don't appear in Lightning Experience.

These page layout elements aren't supported in Lightning Experience:

- Blank spaces
- Expanded lookups

- Mobile cards

  Note: The Twitter card that you see on account, contact, and lead record pages in Lightning Experience isn't the same as the Twitter component available as a mobile card in the page layout editor. The Twitter card in Lightning Experience is a Lightning component. You must have Social Accounts and Contacts enabled for it to appear.

- S-controls
- Sections
- Tags

Note: You can't use the enhanced page layout editor to customize the layout of Lightning Experience record home pages. All users see the same record layout in Lightning Experience, regardless of profile or record type.

Tips for Optimizing Page Layouts for Salesforce1

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** editions except Database.com

Here are some tips and tricks for making your existing page layouts more mobile-friendly.

Page layouts containing dozens of fields and lots of related lists might be manageable when viewing records on a computer screen, but on a small mobile device, viewing that same record can be overwhelming. People accessing information using a mobile device are looking for a quick way to get what they need, and making your users sift through hundreds of fields and related lists just doesn't make sense.

When optimizing a page layout, consider:

- What are the important things to see at a glance?
- What are the important moments for your users when they're working in Salesforce1?
- What actions or processes can you automate so that your users don't have to manually do them?

The Key: Organize and Minimize Fields

- Use sections to organize information logically, putting the most important things at the top of the page so they show up first. Your users don't want to search for fields individually. Organizing similar fields in sections will help your users find what they need. They can then easily scroll down the page to the section they care about.
- For accounts, contacts, and leads, you don't need to put phone or email fields near the top. They're already quickly accessible via the and icons on each record page's action bar.
- You don't need to keep fields in one column, as the page will render dynamically based on the device that's viewing it. A phone will reorder the fields into a single column, and a tablet or desktop will show two columns.
- Put the most important fields into the compact layout—which drives record highlights and record preview cards in Salesforce1—so they're available right up front, and so your mobile users don't have to drill into the record detail. We'll get into compact layouts in more detail soon.
- Keep the number of required fields to a minimum. Setting a field to required means it must appear on the detail page of all page layouts, so consider whether each field is truly required. You might have to convince stakeholders that a field isn't actually necessary for a record to be saved.
- If available in your organization, think about using record types so that fields that aren't common to all records don't have to appear on all records.
- To reduce the number of fields on a screen, consider using default values for new records instead of having the user enter the data.

Manage Mobile Cards in the Enhanced Page Layout Editor

Add expanded lookups, components, and Visualforce pages to the Mobile Cards section of your page layout to have them show up as mobile cards in Salesforce1.

The items you place in the Mobile Cards section don't show up on a record's detail page in the full Salesforce site but only on the record's related information page in Salesforce1.

You can add these types of elements to the Mobile Cards section of a page layout.

- Components
- Expanded lookups
- Visualforce pages
- Canvas apps

Some standard objects have mobile cards that are added for you by default. Customize the page layout to remove them or add additional cards.

1. Access the page layout editor.
2. From the Expanded Lookups, Components, Visualforce Pages, or Canvas Apps categories, drag an element into the Mobile Cards section.
3. Save the page layout.

After saving the layout, the items you added show up immediately in Salesforce1. You may need to refresh to see the changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** editions except Database.com

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

To view page layouts:
- "View Setup and Configuration"

Customize Related Lists

You can customize the buttons, columns displayed, column order, and record sort order of related lists on record detail pages in Salesforce and the Salesforce Customer Portal.

1. Access the page layout editor.
2. Double-click a related list on the layout to edit it. If you are using the enhanced page layout editor, you can also click the wrench icon ( ).

   Note: You can't customize the History related list because it doesn't store data. The History related list links to data stored elsewhere.

3. Select which fields to include in the related list, and define the order in which the fields display. You can include up to 10 fields per related list.

   Note: You can't move the first field of a related list, because it's a unique identifier for the record.

   You can add custom fields of the long text area type to a related list. However, you can't add some standard fields of the long text area type. For example, you can't add the Description field on an Opportunity to a related list.

   The default sort order varies per record. The `Sort By` drop-down isn't available for activities and opportunity products.

   Lookup fields aren't available for display on their corresponding lookup related list. For example, the case lookup field on an account page layout isn't available when editing the cases related list.

4. If desired, select other page layouts to apply your related list customizations to.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except **Database.com**

USER PERMISSIONS

To customize related lists:
- "Customize Application"

Only layouts that include this related list appear in the list. Layouts that include related lists with the same customizations as the current layout had when you opened it are selected by default.

5. Click **+** in the Buttons section to customize which standard and custom buttons are displayed in the related list.

   Note: The custom button must be defined for the object contained in the related list, not the parent object, and the button `Type` must be List Button. For example, to display a custom button on the Contacts related list of an account, define the custom button for contacts, not accounts.

6. If necessary, click **Revert to Defaults** to undo any customizations and use the default Salesforce settings in the related list.
7. Click **OK** to store your customizations. Changes aren't saved until you save the page layout.
8. Select `Overwrite users' customized related lists` to apply the related lists in the page layout to all users, even if they have already customized their display.
9. Save the page layout.

Note: You can enable related list hover links so that record detail pages include links for each related list at the top of the page. Users can hover the mouse over a related list hover link to display the corresponding related list in an interactive overlay that allows users to quickly view and manage the related list items. Users can also click a related list hover link to jump to the content of the related list without scrolling down the page.

SEE ALSO:

Page Layouts and Field-Level Security

Customize Detail Page Buttons

When customizing page layouts, you can control which standard and custom buttons are displayed and the order in which the custom buttons are shown. To customize the buttons on record detail pages:

1. From the management settings for the object whose page layout you want to customize, go to Page Layouts.
2. Click **Edit** next to the page layout you want to customize.
3. Do one of the following.
   - In the original page layout editor, double-click the Detail Page Buttons item in the Button Section.
     - To hide any standard button, deselect the checkbox next to the button name.
     - To add or remove a custom button, select the button in the Available Buttons list, and click **Add** or **Remove**.
     - Sort custom buttons by selecting them and clicking **Up** or **Down**.
     - To undo your customizations and restore default settings, click **Revert to Defaults**.
     - Click **OK** to close the popup when you are done.
   - In the enhanced page layout editor, select the Buttons category on the palette and drag one or more buttons from the palette to the buttons section on the page layout. Standard buttons must go in the standard buttons area, and custom buttons must go in the custom buttons area. To remove a standard or custom button from the page layout, drag the button to the palette.
4. Click **Save** on the page layout.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To customize detail page buttons:

- "Customize Application"

5. Click **Save**.

SEE ALSO:

Override Standard Buttons and Tab Home Pages

Define Custom Buttons and Links

Find Object Management Settings

Customize Page Layouts with the Original Page Layout Editor

The original page layout editor allows you to customize page layouts for detail and edit pages in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal.

Note: Salesforce recommends using the enhanced page layout editor instead of the original page layout editor.

EDITIONS

Available in: Salesforce Classic

Available in all editions

USER PERMISSIONS

To customize page layouts:
- "Customize Application"

1. Access the page layout editor.
   - Edit Page Layouts for Standard Objects
   - Edit Page Layouts for Custom and External Objects
2. If tags are enabled, specify whether personal and public tags should be included in the header section of the page layout. Users can tag a record only if personal or public tags are included here.
   - To add personal or public tags, select Header Items from the View drop-down list and then drag the Personal Tags or Public Tags items to the header section. You can't change the order in which personal and public tags appear when both are in the header section at the same time.
   - To remove tags, drag the Personal Tags and Public Tags items from the header section to the area under the View drop-down list.
3. To customize buttons, double-click **Detail Page Buttons** in the Button section.
   - To hide any standard button, deselect the checkbox next to the button name.
   - To add or remove a custom button, select the button in the Available Buttons list, and click **Add** or **Remove**.
   - To sort custom buttons, select them and click **Up** or **Down**.
   - To undo your customizations and restore default settings, click **Revert to Defaults**.
   - To close the popup, click **OK**.
4. To arrange fields, custom s-controls, Visualforce pages, custom links, and related lists on the layout, select one or more items from the box on the right and drag them to the desired location. You can drag up to 20 s-controls, 20 Visualforce pages, 20 expanded lookups, and 100 related lists onto a page layout. There are no limits on fields and custom links.
   - To select multiple items individually, use CTRL+click.
   - To select multiple items as a group, use SHIFT+click..

   Note:
   - You can add a Visualforce page to a page layout only if the standard controller on the Visualforce page is set to the object for which you are creating the page layout. If you don't have any Visualforce pages with a standard controller set to that object, the Visualforce Pages category doesn't appear in the palette.
   - Items that are not in the page layout are displayed in the scrolling box on the right-hand side.

- Use the legend to determine what fields are required, dependent, controlling, or have other attributes that may affect your page layout decisions.

5. To set which fields are required and read only, select one or more fields and click **Edit Properties**.
   - The field properties of some standard fields cannot be changed. Custom fields can be changed only if they are not universally required fields.
   - Fields marked as read only are always editable by administrators and users with the "Edit Read Only Fields" permission.
   - If you make a picklist field read only, all new records will contain the default value for that picklist.
   - Auto-number fields are always read only.
   - If you mark the opportunity `Probability` field as read only, the `Probability` value will still be updated automatically when a user changes the `Stage` value of an opportunity.
   - In Professional, Enterprise, Unlimited, Performance, and Developer Editions, field-level security settings override any field properties you set here if the field-level security is more restrictive than the page layout setting.
6. To change the properties of an s-control or Visualforce page, double click it and set the following attributes.
   - `Width` sets the horizontal size in pixels or a percent.
   - `Height` sets the vertical size in pixels.
   - `Show scrollbars` determines whether the iFrame in which the s-control displays contains scrollbars when necessary.
   - `Show label` determines whether the page layout includes the `Label` of the custom s-control. Remove the label to display the custom s-control in a wider area.
7. To organize the page using sections, click **Edit** next to an existing page section, or click **Create New Section** to create a new page section.
   - Enter a name for the section. Note that names of some standard page sections cannot be changed.
   - Set whether the section should have one or two columns.
   - Set the order in which users can tab through the items in that section.
   - Set whether the section name should be shown on the detail and edit pages.
8. To customize related lists on the page layout, double-click a related list in the Related List section.
   - To add or remove fields, select one or more fields and use the arrows to add or remove them to the related list columns on the page layout and to define the order in which the related list columns display. You can include up to 10 fields per related list.
   - To select multiple fields individually, use CTRL+click.
   - To select multiple fields as a group, use SHIFT+click.
   - To sort the items in the related list, select a field from the Sort By drop-down list. Items are displayed in ascending order unless you select `Descending`. The default sort order varies per record. The Sort By drop-down is not available for activities and opportunity products.
   - If necessary, select additional page layouts to which your related list customizations will apply.
   - To customize which standard buttons display in the related list, select or deselect the checkbox next to the button name.
   - To customize which custom buttons display in the related list, select the button and click **Add** or **Remove**. To sort custom buttons, select them and click **Up** or **Down**.

     Note: To create a custom button for the related list, see Define Custom Buttons and Links. The custom button must be defined for the object contained in the related list, instead of the parent object, and the button `Type` must be List Button. For example, to display a custom button on the Contacts related list of an account, define the custom button for contacts, not accounts.

Some related lists are not customizable because they link to data rather than store it. You can move your cursor over any related list section to see if it is customizable. Also, lookup fields are not available for display on their corresponding lookup related list. For example, the case lookup field on an account page layout is not available when editing the cases related list.

Note: You can enable related list hover links so that record detail pages include links for each related list at the top of the page. Users can hover the mouse over a related list hover link to display the corresponding related list in an interactive overlay that allows users to quickly view and manage the related list items. Users can also click a related list hover link to jump to the content of the related list without scrolling down the page.

9. To apply the related lists in the page layout to all users, even if they have already customized their display, select `Overwrite users' customized related lists`.
10. To review the page layout, click **Preview**. From the preview in Enterprise, Unlimited, Performance, and Developer Editions, select a profile to see how the pages will look for users with different profiles. Note that most related lists' columns preview without data.
11. Click **Save** to finish. Alternatively, click **Quick Save** to save and continue editing the page layout.

In Professional, Enterprise, Unlimited, Performance, and Developer Editions:.

- To choose which related records display in the Console tab's mini view, click **Mini Console View**.
- To define the mini page layouts of the records that appear in the Console tab's mini view, click **Mini Page Layout**.

Note: You cannot define mini console views or mini page layouts for the Close Case Layout or the Log a Case Page and View Cases Page layouts on the Self-Service Portal.

In Enterprise, Unlimited, Performance, and Developer Editions:

- You can assign page layouts for different profile and record type combinations.
- You can set field-level security to restrict field access further.

SEE ALSO:

Notes on Using the Original Page Layout Editor

Customize Related Lists

Page Layouts

Notes on Using the Original Page Layout Editor

EDITIONS

Available in: Salesforce Classic

Available in all editions

- When customizing page layouts for tasks, you can select the following checkboxes. (These options are unavailable when user control over task assignment notifications is enabled.)
  - `Show Task Email Notification checkbox` displays the `Send Notification Email` checkbox when users create or edit a task.
  - `Select Task Email Notification checkbox by default` selects the `Send Notification Email` checkbox by default when users create or edit a task. Note that a user's personal preference for defaulting the state of the checkbox takes precedence over the organization-wide setting.
- When customizing page layouts for cases, you can select the following checkboxes. (These options are unavailable when user control over task assignment notifications is enabled.)
  - `Show on edit page Case Assignment checkbox` displays the `Assign using active assignment rules` checkbox when users create or edit a case.
  - `Default Case Assignment checkbox` automatically runs case assignment rules.

If both `Show on edit page` and `Select by default` are selected, the assignment checkbox is selected by default, but users can deselect it to override the assignment rule.

  - `Show Case Email Notification checkbox` displays the `Send Notification Email` checkbox when users create or edit a case.
  - `Select Case Email Notification checkbox by default` selects the `Send Notification Email` checkbox by default when users create or edit a case.

- Page layouts for the user object only include custom fields, custom links, s-controls, and Visualforce pages. Tagging, related lists, custom buttons, and standard field customizations are *not* included on page layouts for the user object. Also, field-level security is available only for custom fields on the user object.
- You can define mini page layouts for the user object; however, you cannot add standard fields or related lists. Also, a customized mini page layout won't display in the Agent console.
- Users can import values into a field only if they have read and edit access. Field access is determined by user permissions, page layout assignments, and field-level security settings.
- In Personal, Contact Manager, and Group Editions, page layouts control which fields users can access in related lists, list views, reports, Connect Offline, email and mail merge templates, custom links, and when synchronizing data. In Professional, Enterprise, Unlimited, Performance, and Developer Editions, this access is controlled by field-level security.
- In Professional, Enterprise, Unlimited, Performance, and Developer Edition, individual users can customize which tabs and related lists display for their personal use.
- When editing a person account page layout, if you add `Shipping Address` next to `Billing Address` in the Address Information section, a link will display on the person account edit page that allows you to copy the billing address to the shipping address. Also, an equivalent link appears if you add `Other Address` to the Address Information section.
- Some items can only be moved to certain sections on the page layout. For example, you can drag a custom s-control to any field section on the page layout but not a related list section or button section.
- Create the appropriate buttons before editing your page layout. For example, create an account custom button for the detail page and a contact custom list button before putting them both on an account page layout.
- If you use the original page layout editor to view a page layout that was created in the new page layout editor, the original page layout editor will show any blank spaces you added. You cannot move or add blank spaces in the original page layout editor, but you can remove them by dragging them to the box on the right.

SEE ALSO:

Customize Page Layouts with the Original Page Layout Editor

Compact Layouts

Compact layouts are used in Salesforce1 and Lightning Experience to display a record's key fields at a glance.

In the full Salesforce site, compact layouts determine which fields appear in the Chatter feed item that appears after a user creates a record with a quick action.

Note: To avoid inadvertent sharing of information through the feed, the Task page layout determines the fields displayed in the Chatter feed items for tasks created using a quick action.

In Salesforce1, the first four fields that you assign to a compact layout appear in:

- An object's record highlights area
- Expanded lookup cards on a record's related information page

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions except **Database.com**

In Lightning Experience, the first five fields that you add to a compact layout display in an object's record highlights panel.

Note: If a user doesn't have access to one of the fields that you assign to a compact layout, the next field on the layout is used.

If you don't create custom compact layouts for an object, all the object's record highlight fields, preview cards, and action-related feed items are driven by a read-only, system default compact layout that contains a predefined set of fields. After you create one or more custom compact layouts, you can set one as the primary compact layout for the object. The primary compact layout is then used as the default for that object.

Primary compact layouts determine which fields are shown in Chatter personal digest emails.

If you have record types associated with an object, you can override the primary compact layout assignment and assign specific compact layouts to different record types.

Compact layouts support all field types except:

- text area
- long text area
- rich text area
- multi-select picklist

SEE ALSO:

Create Compact Layouts
Notes on Compact Layouts

Create Compact Layouts

Use compact layouts to customize the fields that display for object records when viewed in Salesforce1 and Lightning Experience.

The first four fields on your compact layout populate the record highlights section at the top of each record view in Salesforce1. The record highlights section in Lightning Experience uses the first five fields on the compact layout.

1. From the management settings for the object that you want to edit, go to Compact Layouts.
2. Create a new compact layout and give it a label.
3. Add up to 10 fields.

   Tip: Put the object's `Name` field first to provide context for your users when they view a record.

4. Sort the fields by selecting them and clicking **Up** or **Down**.

   The order you assign to the fields determines the order in which they display.

5. Save the layout.
6. To set the compact layout as the primary compact layout for the object, click **Compact Layout Assignment**.

Example: This image shows a sample edit compact layout page for the standard Account object:

EDITIONS

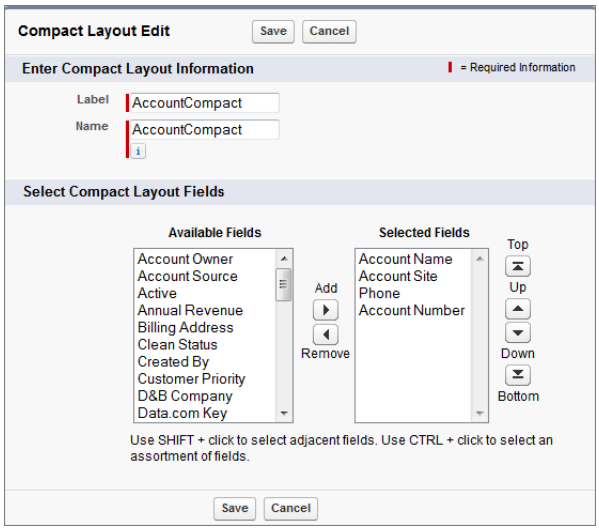
Available in: both Salesforce Classic and Lightning Experience

Available in: **All** editions except **Database.com**

USER PERMISSIONS

To customize compact layouts:
- "Customize Application"

To view compact layouts:
- "View Setup and Configuration"



Here is the detail page for a sample account in the Salesforce1 mobile app. You can see that the account's name, site, phone number, and account number are concisely displayed at the top of the page.

SEE ALSO:

- Assign Compact Layouts to Record Types
- Notes on Compact Layouts
- Compact Layouts

Assign Compact Layouts to Record Types

If you don't create custom compact layouts for an object, all the object's record highlight fields, preview cards, and action-related feed items are driven by a read-only, system default compact layout that contains a predefined set of fields. After you create one or more custom compact layouts, you can set one as the primary compact layout for the object. The primary compact layout is then used as the default for that object. If you have record types associated with an object, you can override the primary compact layout assignment and assign specific compact layouts to different record types.

1. From the management settings for the object that you want to edit, go to Compact Layouts.

   Tip: For Salesforce Knowledge articles, from Setup, enter *Knowledge Article Types* in the Quick Find box, then select **Knowledge Article Types**, click the name of an article type, then scroll down to the Compact Layouts related list.

2. Click **Compact Layout Assignment**.
3. Select a compact layout to use as the primary compact layout for this object.
4. In the Record Type Overrides section, select one or more record types to which you want to assign a compact layout.

   If you don't have record types set for the object, you won't see this section. If you don't set any record type overrides, all record types use the object's primary compact layout by default.

   Some record types in the list might be inactive. You can assign a compact layout to an inactive record type.

5. Select a compact layout from the Compact Layout To Use drop-down list to assign it to the selected cells.
6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Compact layouts are available in: **All** editions except **Database.com**

Record types are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To customize compact layouts:
- "Customize Application"

To view compact layouts:
- "View Setup and Configuration"

SEE ALSO:

Create Compact Layouts
Notes on Compact Layouts
Compact Layouts
Find Object Management Settings

Notes on Compact Layouts

- Changes you make to a compact layout are reflected in both Salesforce1 and Lightning Experience.
- Each record type can have only one compact layout assigned to it. However, a compact layout can be associated with multiple record types.
- Users who don't have access to certain fields in Salesforce won't see them on the compact layout.
- A compact layout must contain at least one field.
- A compact layout can only contain fields from its object, including a formula field that is a cross-object reference to another object.
- Removing a field from a page layout doesn't remove it from the object's compact layout. The two layout types are independent.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions except **Database.com**

- If you change a field on a compact layout to an unsupported type, the field is removed from the compact layout.
- Before you can delete a compact layout that's set as the primary compact layout for the object, you must choose another compact layout to replace it.
- In Salesforce1, tasks automatically show whether a task is open or closed and the due date (depending on a user's access to activity dates). When customizing a task compact layout, you don't need to add these fields to the Selected Fields list.
- Compact layouts aren't available for the Chatter Mobile for BlackBerry or Dashboards for iPad mobile apps.

SEE ALSO:

Create Compact Layouts

Compact Layouts

Select Fields to Display in the Recently Viewed List on Object Home Pages in Lightning Experience

As a Salesforce admin, you can customize the Recently Viewed list that appears on the home page for most standard and custom objects. Choose and order the fields to display so that your users see the information that's most important for your company.

Note: These steps work in Lightning Experience. If you see the App Launcher icon (⁞⁞⁞) on the left side of the navigation bar at the top of your screen, you're in Lightning Experience. If not, you're in Salesforce Classic.

For most list views, your users can select which fields to display and how to order the view columns. However, they can't edit the recent records quick list on object home pages. Only Salesforce admins can select and order the fields to display for the recent records quick list. Admins can't make any other changes to this default list.

EDITIONS

Available in: Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To customize recent records list
- "Customize Application"

1. From Setup, enter *Object Manager* in the Quick Find box, then select **Object Manager**.
2. Click the label name of the object for the Recently Viewed list you want to modify.
3. From the menu of links at the top of the page, click **Search Layouts**.
4. In the far right of the Search Results column, click ▼ and select **Edit**.
5. To add columns to the Recently Viewed list, select one or more fields from Available Fields and click **Add**. To remove columns, select one or more fields from Selected Fields and click **Remove**.
6. Order columns by selecting one or more fields from Selected Fields and clicking **Up** or **Down**.
7. Click **Save**.

Example: Your users collaborate on opportunities. To make it easy to see who worked on a recent opportunity last, select **Last Modified By** from the Available Fields list. Click **Add** to move it to Selected Fields. Now this information appears on the Recently Viewed list on the Opportunities home page.

Customize Search Layouts

Customize which fields display for users in search results, search filter fields, lookup dialogs, the recent records lists on tab home pages in Salesforce Classic, and in lookup phone dialogs for Salesforce CRM Call Center.

You can specify a different set of fields to show in each search layout. The settings apply to all users in your organization and Salesforce Customer Portal.

You can also customize which buttons display in custom list views and search results. You can hide a standard list view button or display a custom button. Standard buttons aren't available on search result layouts. To display a custom button, create the custom button, giving it the `List Button Display Type`.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To change search layouts:
- "Customize Application"

1. From the management settings for the appropriate object, go to Search Layouts.
2. Click **Edit** next to the layout you want to customize. Specify a different set of items to display. You can edit search results, lookup dialogs, recent records lists on tab home pages in Salesforce Classic, lookup phone dialogs, list views, and search filter fields.
3. For list view and search results layouts, select the standard or custom buttons you want to display. To hide a standard button on the list view, deselect it. Standard buttons aren't available on search result layouts.
4. Move fields between Available Fields and Selected Fields.
   - To customize which fields display in the layout, select one or more fields and click **Add** or **Remove**.
   - To sort fields in the layout, select one or more fields in Selected Fields and click **Up** or **Down**.
   - To select multiple fields individually, use CTRL+click.
   - To select multiple fields as a group, use SHIFT+click.
   - To customize which items display in the layout, select the item and click **Add** or **Remove**.
   - To sort items in the layout, select the item and click **Up** or **Down**.

   Note: When editing a search results layout for an object, you can select the **Override the search result column customizations for all users** checkbox. If selected, all user column customizations within your organization are overwritten and restored to the organization-wide default settings.
5. Click **Save**.

SEE ALSO:

Page Layouts

Notes on Search Layouts

Notes on Search Layouts

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To change search layouts:
- "Customize Application"

- Search layouts don't apply to Salesforce CRM Content.
- Search layouts don't apply to campaign members, opportunity teams, or account teams.
- The search layout doesn't control which fields are searched for keyword matches. The list of fields searched is the same across Salesforce.
- You can add up to 10 fields to each search layout.
- You can't remove unique identifying fields, such as `Account Name` or `Case Number`, from the search layouts. These fields must be listed first in the order of fields in the search layout.
- You can 't add long text fields such as `Description`, `Solution Details`, or custom long text area fields to search layouts.
- All fields are available to be added to the search layout even if some fields are normally hidden for the user customizing the search layout.
- For Professional, Enterprise, Unlimited, Performance, and Developer Edition organizations, search layouts don't override field-level security. If a field is included in the search layout but hidden for some users via field-level security, those users do not see that field in their search results.
- For Personal, Contact Manager, Group, and Professional Edition organizations, search layouts override page layout settings. If a field is included in the search layout but hidden in the page layout, that field will be visible in search results.
- The search results layouts for leads, accounts, contacts, and opportunities also apply to the search results displayed when finding duplicate leads.
- Formula fields are not available in search result layouts.
- Don't remove the `Phone` field from any lookup phone dialogs search layout. If you do, users can't use the directory search results to enter a phone number into a SoftPhone dial pad.
- To add a custom button to a list view or search layout, create the custom button for a standard or custom object, giving it the **List Button** `Display Type`. The custom button will be available in the list view and search result layouts for that object.

Tip: In account search results, you can visually differentiate business accounts from person accounts by adding the `Is Person Account` field, which displays as the person account icon ( ).

You may specify the `Is Person Account` field as the first column in account search layouts. Otherwise, `Account Name` must be the first column.

Custom Tabs

Custom tabs let you display custom object data or other web content in Salesforce. When you add a custom tab to an app in Salesforce Classic, it displays as a tab. When you add a custom tab to an app in Lightning Experience, it displays as an item in the app's navigation bar.

Custom tabs display custom object data or other web content embedded in the app. You can create any of these types of custom tabs.

Custom Object Tabs
Custom object tabs (available only at an app level and not on subtab apps) display the data of your custom object. Custom object tabs look and function just like standard tabs.

Web Tabs
Custom web tabs display any external web-based application or web page. You can design web tabs to include the sidebar or span across the entire page without the sidebar.

Visualforce Tabs
Visualforce tabs display data from a Visualforce page. Visualforce tabs look and function just like standard tabs.

Lightning Component Tabs
Lightning component tabs make Lightning components available in the Salesforce1 mobile app (via the navigation list) or in Lightning Experience. Lightning components aren't supported in Salesforce Classic.

Note: To expose Lightning components via Lightning component tabs, you must enable and deploy My Domain.

Lightning Page Tabs
Lightning Page tabs let you add "App Page" Lightning Pages to the Salesforce1 navigation list and the Lightning Experience navigation bar.

In Salesforce Classic, Lightning Page tabs don't show up on the All Tabs page when you click the plus icon ( ) that appears to the right of your current tabs. Nor do Lightning Page tabs show up in the Available Tabs list when you customize the tabs for your apps.

Subtab apps support only web tabs and Visualforce tabs.

Delegated administrators who can manage specified custom objects can also create and customize tabs for those custom objects.

SEE ALSO:

Create Custom Apps for Salesforce Classic

What is a Subtab App?

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Lightning Page Tabs available in: **All** Editions except **Database.com**

USER PERMISSIONS

To create and edit custom tabs:
- "Customize Application"

Create Web Tabs

Build web tabs so that your users can use your web applications or other websites from within the application.

1. From Setup, enter *Tabs* in the Quick Find box, then select **Tabs**.
2. Click **New** in the Web Tabs related list.
3. Choose a layout for the new tab. The full page width spans across the entire page without the sidebar, while the column style allows users to view the sidebar.
4. Click **Next**.
5. Enter a label to display on the tab.
6. Click the Tab Style lookup icon to display the Tab Style Selector.

   If a tab style is already in use, a number enclosed in brackets [] appears next to the tab style name. Hover your mouse over the style name to view the tabs that use the style. Click Hide styles which are used on other tabs to filter this list.
7. Click a tab style to select the color scheme and icon for the custom tab.

   Optionally, click **Create your own style** on the Tab Style Selector dialog to create a custom tab style if your org has access to the Documents tab. To create your own tab style:

   a. Click the **Color** lookup icon to display the color selection dialog and click a color to select it.

   b. Click **Insert an Image**, select the document folder, and select the image you want to use.

      Alternatively, click **Search in Documents**, enter a search term, and click **Go!** to find a document file name that includes your search term.

      Note: This dialog only lists files in document folders that are under 20 KB and have the Externally Available checkbox selected in the document property settings. If the document used for the icon is later deleted, Salesforce replaces it with a default multicolor block icon ( ).

   c. Select a file and click **OK**. The New Custom Tab wizard reappears.
8. Change the content frame height if necessary.
9. Optionally, select Salesforce Classic Mobile Ready to indicate that the web page displays and functions properly in the Salesforce Classic Mobile app.

   Selecting this checkbox adds the tab to the list of available tabs for your Salesforce Classic Mobile configurations. Before mobilizing a web tab, review the Salesforce Classic Mobile tab considerations to ensure that the pages in your web tabs are compatible with mobile browsers.

   Note: The Salesforce Classic Mobile Ready checkbox is only visible if Salesforce Classic Mobile is enabled for your organization.
10. Optionally, choose a custom link to use as the introductory splash page when users initially click the tab. Don't use a splash page if you plan to mobilize the web tab, as splash pages don't display in the mobile application.
11. Enter a description of the tab, if desired, and click **Next**.
12. Enter the URL or choose the custom s-control that you want to display in the tab. Optionally, copy and paste any merge fields for data that you want dynamically replaced in the link.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Lightning Page Tabs available in: **All** Editions except **Database.com**

USER PERMISSIONS

To create and edit custom tabs:

- "Customize Application"

Note: Only User, organization, and API merge fields are supported for web tabs.

Click **Preview Web Tab** to display your web tab.

13. For a URL, choose an encoding setting and click **Next**.
14. Add the web tab to the appropriate profiles. Choose Default On, Default Off, or Tab Hidden to determine whether the custom tab is visible to users with that profile. You can change this setting later.
15. Click **Next**.
16. Specify the custom apps that should include the new tab.
17. Select **Append tab to users' existing personal customizations** to apply the tab visibility settings to all users.
18. Click **Save**.

Create a Custom Object Tab

Define a tab to display the data stored in your custom object records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Lightning Page Tabs available in: **All** Editions except **Database.com**

USER PERMISSIONS

To create and edit custom tabs:
- "Customize Application"

1. From Setup, enter *Tabs* in the Quick Find box, then select **Tabs**.
2. Click **New** in the Custom Object Tabs related list.
3. Select the custom object to display in the custom tab. If you have not already created the custom object, click **create a new custom object now** and follow the instructions in Define a Custom Object on page 83.

   The label of the new tab is the same as the plural version of the custom object label.
4. Click the Tab Style lookup icon to display the Tab Style Selector.

   If a tab style is already in use, a number enclosed in brackets [] appears next to the tab style name. Hover your mouse over the style name to view the tabs that use the style. Click Hide styles which are used on other tabs to filter this list.
5. Click a tab style to select the color scheme and icon for the custom tab.

   Optionally, click **Create your own style** on the Tab Style Selector dialog to create a custom tab style if your org has access to the Documents tab. To create your own tab style:

   a. Click the **Color** lookup icon to display the color selection dialog and click a color to select it.

   b. Click **Insert an Image**, select the document folder, and select the image you want to use.

      Alternatively, click **Search in Documents**, enter a search term, and click **Go!** to find a document file name that includes your search term.

      Note: This dialog only lists files in document folders that are under 20 KB and have the Externally Available checkbox selected in the document property settings. If the document used for the icon is later deleted, Salesforce replaces it with a default multicolor block icon ( ).

   c. Select a file and click **OK**. The New Custom Tab wizard reappears.
6. Optionally, choose a custom link to use as the introductory splash page when users initially click the tab.
7. Enter a description of the tab, if desired, and click **Next**.

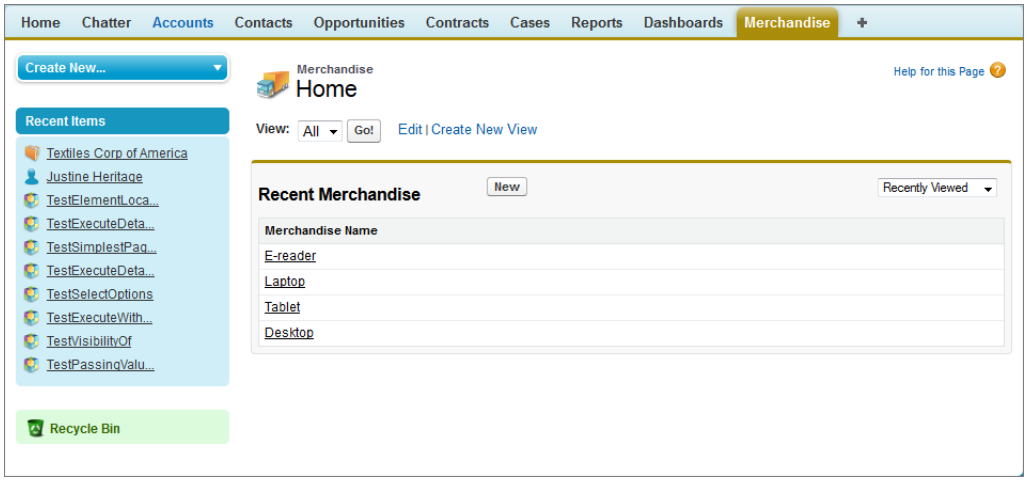
8. Choose the user profiles for which the new custom tab will be available.

   For Professional Edition users and Salesforce Platform One license users, tab visibility is set to Default On.

   Note: Custom object records are searchable in the Salesforce user interface only if the custom object is associated with a custom tab. Users aren't required to add the tab for display.

9. Specify the custom apps that should include the new tab.
10. Select **Append tab to users' existing personal customizations** to add the tab to your users' customized display settings if they have customized their personal display.
11. Click **Save**.

    Depending on the visibility settings you selected, you'll see the tab right away:



SEE ALSO:

Custom Tabs

Create Lightning Page Tabs

Before you can include an App Page type of Lightning Page in the Salesforce1 navigation list or in a Lightning app, you must create a custom tab for it.

When you first activate an app page in the Lightning App Builder, a tab is created for the page as part of the activation process. You can also create a tab for the page manually in Setup before you activate it.

Lightning Page tabs only facilitate inclusion of the Lightning Page in Salesforce1 and Lightning Experience.

You can create a custom tab only for an App Page type of Lightning Page.

1. From Setup, enter *Tabs* in the `Quick Find` box, then select **Tabs**.
2. Click **New** in the Lightning Page Tabs related list.
3. Choose a Lightning Page for the tab.
4. Enter a label.

EDITIONS

Available in: Salesforce Classic

Available in: **All** editions except **Database.com**

USER PERMISSIONS

To create and edit custom tabs:

- "Customize Application"

This text is used as the display name for the Lightning Page.

5. Select a tab style to set a color scheme and icon for the Lightning Page tab.

   Optionally, click **Create your own style** on the Tab Style Selector dialog to create a custom tab style if your org has access to the Documents tab. To create your own tab style:

   a. Click the **Color** lookup icon to display the color selection dialog and click a color to select it.

   b. Click **Insert an Image**, select the document folder, and select the image you want to use.

      Alternatively, click **Search in Documents**, enter a search term, and click **Go!** to find a document file name that includes your search term.

      Note: This dialog only lists files in document folders that are under 20 KB and have the Externally Available checkbox selected in the document property settings. If the document used for the icon is later deleted, Salesforce replaces it with a default multicolor block icon ( ).

   c. Select a file and click **OK**. The New Custom Tab wizard reappears.

6. Enter a description of the tab, if desired, and click **Next**.

7. Choose the user profiles for which the new custom tab will be available.

   Note: In Salesforce Classic, the Default On and Default Off options for Lightning Page tabs don't work the same way as for other custom tabs. The Lightning Page menu item appears for the selected profiles in Salesforce1 whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning Page for the selected profiles.

8. Click **Save**.

Custom Tab Considerations

When working with custom tabs, keep these considerations in mind.

- To enable Visualforce or web tabs for use in the Salesforce Classic Mobile app, edit the properties of the tabs and select the **Salesforce Classic Mobile Ready** checkbox.
- The custom tabs limit is a fixed number based on edition and can't be increased. For more information, contact Salesforce.
- If you choose Default On or Default Off when setting custom object tab visibility, an option is added to the `Create New` drop-down list in the Salesforce Classic sidebar so that users with the "Create" permission can quickly create a record. For example, if the custom object displayed in the custom tab is named Expenses, an **Expense** option appears in this list.
- Custom object records are searchable in the Salesforce user interface only if the custom object is associated with a custom tab. Users aren't required to add the tab for display.
- A Web tab or custom link could display a blank page if the embedded site:
  - Has been set to deny the loading of its content in a frame.
  - Has been set to allow the loading of its content in a frame only if the same site is delivering the content.
  - Contains a mix of secure and unsecure content, and the user's browser has been configured to block mixed active content.

  To resolve this issue, try these workarounds.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce Tabs available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Lightning Page Tabs available in: **All** Editions except **Database.com**

  - Set your custom link to either open in a new window or display in the existing window without the sidebar or header.
  - Move the URL from a Web tab into a custom link instead, and set the URL to either open in a new window or display in the existing window without the sidebar or header.
  - If the site you're embedding has an HTTP prefix and mixed active content, try changing the prefix to HTTPS. If the embedded site has a valid security certificate and it hasn't blocked itself from being displayed in frames, using HTTPS as the prefix allows the site to display.
- In Salesforce Classic, Lightning Page tabs don't show up on the All Tabs page when you click the plus icon ( ) that appears to the right of your current tabs. Nor do Lightning Page tabs show up in the Available Tabs list when you customize the tabs for your apps.
- In Salesforce Classic, the Default On and Default Off options for Lightning Page tabs don't work the same way as for other custom tabs. The Lightning Page menu item appears for the selected profiles in Salesforce1 whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning Page for the selected profiles.

Custom Help

The **Help & Training** link at the top of every page opens the Salesforce Help & Training window which includes online help topics, solutions, and recommended training classes. Additionally, the **Help for this Page** link on any page opens a context-sensitive online help topic that describes that page.

Salesforce custom help functionality allows you to augment these standard help features with information on using fields and functionality unique to your organization or the Force.com AppExchange app you are developing.

Custom help allows you to:

- Override the standard Salesforce context-sensitive online help topics for your custom objects using object-level help
- Add field-level help that displays when users hover their mouse over a field

EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

SEE ALSO:

Getting Started with Field-Level Help

Getting Started with Object-Level Help

Getting Started with Field-Level Help

Field-level help lets you provide help text detailing the purpose and function of any standard or custom field. You can define custom help text for your organization's fields to provide users with a helpful description for any field on all detail and edit pages where that field displays. Users can view the field-level help text by hovering over the Info icon next to the field.

Before you begin defining field-level help, review these implementation tips and best practices.

Implementation Tips

- Field-level help is enabled by default for all editions.
- Field-level help is not available for some standard fields, including fields on the User object, system read only fields, auto-number fields, multi-currency fields, Ideas fields, and Community fields.
- The help text for a field is automatically added to a package when you add the associated field to any Force.com AppExchange package.
- In a managed package, the help text is locked to the developer, giving installers full capabilities to change it.

Best Practices

- Because your custom help text displays on both edit and detail pages, avoid instructions for entering data. Instead, construct help text that defines the field's purpose, such as:

  The maximum discount allowed for this account.

- Provide information in your help text about the attributes of the field, such as:

  A detailed description of the purpose for the expense report. Up to 32 KB of data are allowed. Only the first 255 characters display in reports.

- Provide examples in your help text that help users understand the field's meaning clearly, such as:

  The four-digit promotional code used to determine the amount charged to the customer, for example, 4PLT (for level-four platinum pricing).

- If your organization uses more than one language, provide translations for your `Help Text` using the Translation Workbench.

SEE ALSO:

Custom Help

Define Field-Level Help

Define Field-Level Help

Field-level help lets you provide help text detailing the purpose and function of any standard or custom field. You can define custom help text for your organization's fields to provide users with a helpful description for any field on all detail and edit pages where that field displays. Users can view the field-level help text by hovering over the Info icon next to the field.

1. From the management settings for the field's object, go to Fields.
2. Click **Edit** next to the field.
3. In the `Help Text` field, enter the text you want displayed when a user hovers the mouse over the Info icon that appears next to the field on a detail or edit page. You can enter up to 255 characters.
4. Click **Save**.

Note: If the object you're modifying is exposed in a community, field-level help is visible to community members, including unlicensed users, partners, and customers. Make sure the information you provide in field-level help accounts for all audiences and doesn't contain business-sensitive information.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except **Database.com**

USER PERMISSIONS

To define or change field-level help:

- "Customize Application"

SEE ALSO:

Custom Help

Getting Started with Field-Level Help

Find Object Management Settings

Getting Started with Object-Level Help

Help your users by providing object-level help for your custom and external objects. This way, when your users click the **Help for this Page** link on your custom object, they'll find useful information that's relevant to your custom object.

Note: If you don't create object-level help, the **Help for this Page** link provides information about standard objects that won't be relevant to your custom object.

You can override the **Help for this Page** links for a custom object or external object with help content contained in a Visualforce page. But don't worry! You don't have to learn Visualforce to add help content to your custom objects.

When you add custom help to a custom or external object, the **Help for this Page** link on those object pages displays your custom help instead of generic help. Your users can access the custom help content from the object home (overview), detail, and edit pages, list views, and related lists.

You can create object-level help for all custom objects and external objects in the Salesforce editions that support custom and external objects. For more information, see *Salesforce Limits*.

SEE ALSO:

Custom Help

Define Object-Level Help

Object-Level Help Best Practices and Considerations

Define Object-Level Help

Object-level help overrides the **Help for this Page** links for a custom object or external object with your own help content contained in a Visualforce page. To make object-level help available to all your users, create a Visualforce page that contains your help content. Then edit the custom or external object definition to reference the page. Object-level help becomes available to all your users instantly.

1. Create a Visualforce page that contains your help content.
2. Edit the custom object definition or external object definition that uses the custom help when users click the **Help for this Page** link.
3. For `Context-Sensitive Help Setting`, select **Open a window using a Visualforce page**.
4. Select the Visualforce page that contains your help content.
5. Click **Save**.

SEE ALSO:

Getting Started with Object-Level Help

Custom Help

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom objects are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To define or change object-level help:

- "Customize Application"

Create a Custom Object Help Page with Static Content

If you know HTML, it's easy to add help to your custom objects by writing the content in HTML and saving it in a Visualforce page. No need to learn Visualforce. Just use the template that we provide.

1. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
2. Click **New**.
   The Visualforce page editor opens with a new page.
3. Complete the following fields.

Field	Description
Label	The human-friendly name of the page used to identify the page in Setup tools. Tip: It's a great idea to have a naming convention for your custom help pages. For example, start all custom help pages with "Help_" and then the object name.
Name	The API name for the page. You can use the auto-filled value.
Description	An optional description of the page.
Available for Salesforce mobile apps and Lightning Pages	Select this option if your custom object is available in Salesforce1.

4. Click **Quick Save**.
5. In the **Visualforce Markup** tab code editor, select the default code and delete it.
6. Paste the following help template code into the code editor.

```
<apex:page showHeader="false">

    <!-- Add your help styles -->

    <h1>Help for {YourObjectName} Object</h1>

    <p>Your custom help message here.</p>


</apex:page>
```

7. Click **Quick Save**.
8. Edit the template to add your help content.
   To add formatting to your page, use HTML markup. You can also use Visualforce markup if you know it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom objects are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Visualforce is available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To define or change object-level help:
- "Customize Application"

To create or edit Visualforce pages:
- "Customize Application"

9. Click **Save**.

You can now add this page as custom help. When users click **Help for this Page**, they see this page in the Help & Training window.

SEE ALSO:

- Define Object-Level Help
- Create Custom Object Help with a PDF File
- Object-Level Help Best Practices and Considerations

Create Custom Object Help with a PDF File

Add help to your custom objects by creating Visualforce pages that redirect to PDF help files or a URL. No need to learn Visualforce. Just use the template that we provide.

You can write your help content in an authoring tool such as Microsoft Word and convert it to a PDF file.

1. Upload your PDF file as a static resource.
   Users see this PDF file when they request help with the custom object.
2. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
3. Click **New**.
   The Visualforce page editor opens with a new page.
4. Complete the following fields.

Field	Description
Label	The human-friendly name of the page used to identify the page in Setup tools. Tip: It's a great idea to have a naming convention for your custom help pages. For example, start all custom help pages with "Help_" and then the object name.
Name	The API name for the page. You can use the auto-filled value.
Description	An optional description of the page.
Available for Salesforce mobile apps and Lightning Pages	Select this option if your custom object is available in Salesforce1.

5. Click **Quick Save**.
6. In the **Visualforce Markup** tab code editor, select the default code and delete it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom objects are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Visualforce is available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To define or change object-level help:
- "Customize Application"

To create or edit Visualforce pages:
- "Customize Application"

7. Paste the following help template code into the code editor.

```
<apex:page showHeader="false" action="{! URLFOR($Resource.YourCustomHelpResource) }">

    <!-- This page redirects to the URL in the action attribute above -->

    <p>Redirecting to help content...</p>

</apex:page>
```

8. Replace `YourCustomHelpResource` in the action attribute with the name of the static resource that you uploaded.
9. Click **Save**.

You can now add this page as help. When users click **Help for this Page**, they're redirected to the resource you set in the action attribute.

Note: The user's browser controls the behavior of a PDF link, not your Visualforce page. The PDF content might display in the browser or be downloaded as a PDF file.

SEE ALSO:

Define Object-Level Help
Create a Custom Object Help Page with Static Content
Object-Level Help Best Practices and Considerations

Object-Level Help Best Practices and Considerations

Before defining object-level help text for your custom or external objects, review these best practices and implementation considerations.

EDITIONS

Available in: Salesforce Classic

Custom objects are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Visualforce is available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Best Practices

- The window that displays your object-level help has the same height and width dimensions as the standard Salesforce Help & Training window. To increase the usability of your help content, size and style your content appropriately.
- To give your custom help a professional tone using Salesforce terminology, follow the *Salesforce Style Guide for Documentation and User Interface Text*.
- Your Visualforce help content pages can use merge fields or other functions to make the experience more personalized. For example, you can design the help to add the user's name when the help page is displayed.

Advanced Implementation Considerations

- Create custom help Visualforce pages without a controller, or use a custom controller. You can't use a standard controller.
- If you have defined object-level help for an object that you add to a Force.com AppExchange package, Salesforce adds the Visualforce page or static resource referenced in your `Context-Sensitive Help Settings` for that object.

- In managed packages, object-level help is locked to the developer, giving installers the ability to change it if needed.

SEE ALSO:

Define Object-Level Help

Getting Started with Object-Level Help

Replace Built-in Salesforce Help with Custom Help

If you rename standard tabs, objects, fields, and other related user interface labels, you can also replace the built-in Salesforce Help with documentation that's customized specifically for your users. To replace the built-in help, simply provide a URL to your custom help.

Users can view this URL whenever they click on any context-sensitive help link on an end-user page or within their personal settings. After you replace the help, the **Help & Training** link at the very top of every page and all Setup pages will continue to display the Salesforce Help.

1. From Setup, enter *Help Settings* in the Quick Find box, then select **Help Settings**.
2. Enter the complete URL for your help file that you would like to replace the Salesforce Help.
3. Click **Save**.

Considerations for Replacing the Salesforce Help

- When you replace the Salesforce Help with your own help file, the **Help & Training** link still displays Salesforce Help. However, other than within Setup, the **Help for this Page** links on all pages are no longer context-sensitive. That is, your help file will open at the same place regardless of which page the user is viewing when they click the link.
- You can make your help context-sensitive by taking advantage of the context-specific parameters that are passed with each help link. For example, the help link from the Opportunities tab home page is constructed as follows (without any line breaks):

  ```
  http://your_help_file.com?loc=help&amp;body=%2Fhelp%2Fdoc%2Fen%2Fhelp2.jsp
  &target=opp_overview.htm&section=Opportunities
  ```

  The values of the `target` and `section` parameters are unique for every page within the application. You can parse these parameters to display context-sensitive help content for your users.

- If your online help file is compatible with the Web browser on your users' mobile devices, you can contact Salesforce to enable a custom URL for the online help in Salesforce Classic Mobile. If you enable a mobile help URL, the **Help** link in the mobile application launches the custom help URL instead of the Salesforce Classic Mobile online help. Don't deploy customized mobile help without testing it thoroughly on all the device models carried by your users. Even if all of your users have the same type of mobile device, the experience can vary widely depending on which operating system version is installed on the device.

Using Record Types

Record Types

Customize business processes, picklist values, and page layouts for multiple groups of users.

Record types let you offer different business processes, picklist values, and page layouts to different users. Create record types for various reasons, like for:

- Opportunities to differentiate your regular sales deals from your professional services engagements and offer different picklist values for each.
- Cases to display different page layouts for your customer support cases versus your billing cases.

Example: Here's an example of how record types can work in your org. Let's say you have two sales divisions, hardware and consulting, and only your consulting division receives leads through seminars. You can choose to display the Seminar contact lead source for the consulting division only.

Step 1: Manage master picklists

Define a list of contact `Lead Source` picklist values that contains all of the values used by both the Hardware and Consulting divisions, including Seminar.

Step 2: Create record types

Create two contact record types: one called Hardware and another called Consulting. This step includes adding master picklist values to the record types.

Step 3: Add record types to profiles

Add the Hardware record type to the profiles for all users in the hardware sales division. Add the Consulting record type to the profiles of all users in the consulting sales division.

Step 4: Set personal options for record types

Allow users of both the hardware and consulting sales divisions to bypass the prompt that asks them to select a record type when creating a new contact. If you have users that create contact records for both sales divisions, they can customize their personal settings to always prompt them to select a record type.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change record types:

- "Customize Application"

SEE ALSO:

Create Record Types

Considerations for Creating and Updating Record Types and Picklists

Considerations for Creating and Updating Record Types and Picklists

Keep these considerations in mind when working with record types and business process picklists.

- Before creating record types, include all of the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.
- The master picklist is independent of all record types and business processes. If you add a picklist value to the master picklist, you must manually include the new value in the appropriate record types. If you remove a picklist value from the master, it is no longer available when creating new records, but records assigned to that value are unchanged.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- When you create a new record type without cloning an existing one, the new record type automatically includes the master picklist values for both standard and custom picklists. You can then customize the picklist values for the record type.
- A user can be associated with several record types. For example, a user who creates marketing campaigns for both U.S. and European divisions can have both U.S. and European campaign record types available when creating new campaigns.
- When creating and editing record types for accounts, opportunities, cases, contacts, or custom objects, check for criteria-based sharing rules that use existing record types as criteria. A record type change may affect the number of records that the rule shares. For example, let's say you have a record type named "Service," and you created a criteria-based sharing rule that shares all Service record types with your service team. If you create another record type named "Support" and you want these records shared with your service team, update the sharing rule to include Support record types in the criteria.
- Renaming a record type doesn't change the list of values included in it.
- Deactivating a record type doesn't remove it from any user profiles or permission sets.
- Deleting a record type also deletes the related sales path.
- Deleting campaign member record types updates the `Campaign Member Type` field on campaign and campaign member records.
- Person accounts are account records to which a special kind of record type has been assigned. These record types are called *person account record types*. Person account record types allow contact fields to be available on the account and allow the account to be used as if it were a contact. A default person account record type named "Person Account" is automatically created when person accounts are enabled for your org. You can change the name of this record type, and you can create additional person account record types.
- When users convert, clone, or create records, these special considerations apply.
  - When a user converts a lead, the new account, contact, and opportunity records automatically use the default record type for the owner of the new records.
  - When a user clones a record, the new record has the record type of the cloned record. If the user's profile doesn't have access to the record type of the cloned record, the new record adopts the user's default record type.
  - When a user creates a case or lead and applies assignment rules, the new record can keep the creator's default record type or take the record type of the assignee, depending on the case and lead settings specified by the administrator.

SEE ALSO:

Record Types

Create Record Types

Limitations for Creating and Updating Record Types and Picklists

Create Record Types

Tip: Before creating record types, include all of the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.

1. From the management settings for the appropriate object, go to Record Types.
2. Click **New**.
3. Choose `Master` from the `Existing Record Type` drop-down list to copy all available picklist values, or choose an existing record type to clone its picklist values.

   Note: When you create a new record type without cloning an existing one, the new record type automatically includes the master picklist values for both standard and custom picklists. You can then customize the picklist values for the record type.

4. Enter a `Record Type Label` that's unique within the object.
5. Enter a `Record Type Name`. The Record Type Name refers to the component when using the Web services API and prevents naming conflicts on package installation in managed packages.
6. For opportunity, case, lead, and solution record types, select a business process to associate with the record type.
7. Enter a description.
8. Select `Active` to activate the record type.
9. Select `Enable for Profile` next to a profile to make the record type available to users with that profile. Select the checkbox in the header row to enable it for all profiles.

   Tip: If each profile is associated with a single record type, users will never be prompted to select a record type when creating new records.

10. For enabled profiles, select `Make Default` to make it the default record type for users of that profile. Select the checkbox in the header row to make it the default for all profiles.
11. Click **Next**.
12. Choose a page layout option to determine what page layout displays for records with this record type:
    - To apply a single page layout for all profiles, select `Apply one layout to all profiles` and choose the page layout from the drop-down list.
    - To apply different page layouts based on user profiles, select `Apply a different layout for each profile` and choose a page layout for each profile.
13. Click **Save** to edit the values of the standard and custom picklists available for the record type, or click **Save and New** to create another record type.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change record types:
- "Customize Application"

SEE ALSO:

Record Types

Considerations for Creating and Updating Record Types and Picklists

Limitations for Creating and Updating Record Types and Picklists

Edit Picklists for Record Types and Business Processes

Customize the values in record type or business process picklists based on your organization's unique needs.

1. Select a record type or business process and click **Edit** next to the picklist field to change its values.
2. Add or remove values as needed. Users can choose from these values when creating or editing records.
3. Optionally, choose a default picklist value. Some picklists require a default value. The default value in a dependent field is ignored.
4. Click **Save**.

SEE ALSO:

Custom Fields

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Record types are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Business processes are available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change record types:
- "Customize Application"

To create or change business processes:
- "Customize Application"

Limitations for Creating and Updating Record Types and Picklists

Keep these limitations in mind when working with record types and business process picklists.

- These special picklist fields aren't available for record types because they are used exclusively for sales processes, lead processes, support processes, and solution processes:
  - Opportunity `Stage`
  - Case `Status`
  - Solution `Status`
  - `Lead Status`

  You can use these fields to provide different picklist values for different record types by assigning a different process to each record type.
- These campaign member picklists aren't available for record types:
  - `Status`
  - `Salutation`
  - `Lead Source`
- You can't edit or delete a record type for an object if the object is referenced in Apex.
- You can't deactivate a record type if it is in use by an email routing address for Email-to-Case or On-Demand Email-to-Case.
- Record types can only be assigned to campaign members using the `Campaign Member Type` field on new or existing campaigns. To assign record types to campaign members, add the `Campaign Member Type` field to the campaign page

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

layout. You must have the `Marketing User` user permission to change the campaign member type. You can also add a read-only `Campaign Member Type` field to the campaign members page layout.

- We recommend creating no more than 200 record types. While there is no limit, orgs may have difficulty managing their record types if they exceed 200.

SEE ALSO:

Create Record Types

Considerations for Creating and Updating Record Types and Picklists

Managing Multiple Business Processes

Use multiple business processes to display different picklist values for users based on their profile. Multiple business processes allow you to track separate sales, support, and lead lifecycles.

Sales Processes
Create different sales processes that include some or all of the picklist values available for the opportunity `Stage` field.

Lead Processes
Create different lead processes that include some or all of the picklist values available for the `Lead Status` field.

Support Processes
Create different support processes that include some or all of the picklist values available for the case `Status` field.

Solution Processes
Create different solution processes that include some or all of the picklist values available for the `Status` field.

After creating a sales, support, lead, or solution process, assign the process to a record type. The record type determines the user profiles that are associated with the business process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change business processes:
- "Customize Application"

To view a list of business processes, from Setup, enter *`Processes`* in the `Quick Find` box, then select the appropriate link.

SEE ALSO:

Edit Picklists for Record Types and Business Processes

Administrator tip sheet: Tips & Hints for Multiple Business Processes

Create Multiple Business Processes

Follow these steps to create sales processes, support processes, lead processes, and solution processes.

1. From Setup, enter *Processes* in the Quick Find box, then select the appropriate link.
2. Click **New**.
3. Choose an existing process to copy its picklist values into the new process. Select Master to copy all available picklist values.
4. Enter a name and description for the new process. The name must be unique within the tab.
5. Click **Save**.

   All of the available values in the picklist are displayed. Choose the values that you would like included in the new business process.

Next, add the new business process to a record type, and then make the record type available to users based on profile.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change business processes:
- "Customize Application"

Translation Workbench

If your org has multiple languages enabled, use Translation Workbench to maintain your translated labels in your org. You can manage translated values for any Salesforce supported language.

Note: Standard objects aren't available in the Translation Workbench. Use the rename tabs and labels interface for standard object translation.

If a customized component doesn't have a translated value, the component uses the org's default language. When you deactivate a language, all translations for that language are still available in the Translation Workbench, but users with that language selected see the org's default language values.

Translation Workbench isn't available for single-language orgs. If you aren't sure whether you have a single-language or multi-language organization, contact Salesforce.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Enabling the Translation Workbench makes these changes to your Salesforce org.

- You must edit picklist values individually. You can't mass-edit existing picklist values, but you can still mass-add new values.
- When picklist values are sorted alphabetically, the values are alphabetical by the org's default language.
- Reports have a Filter Language drop-down list in the Filters pane of the report builder. Selecting a language filters on translated strings for any filter criteria that use the starts with, contains, or does not contain operator.
- Import files have a Language drop-down list, and all records and values within the import file must be in the selected language.
- Web-to-Lead and Web-to-Case have a Language drop-down list before you generate the HTML.

IN THIS SECTION:

Enable and Disable the Translation Workbench

The Translation Workbench lets you specify languages you want to translate, assign translators to languages, create translations for customizations you've made to your Salesforce organization, and override labels and translations from managed packages. Everything from custom picklist values to custom fields can be translated so your global users can use all of Salesforce in their language.

Add Translated Languages and Translators

Add supported languages, activate or deactivate a language, assign translators for that language, and enter translated terms for customizations you've made in your org's default language.

Translate Terms

Override Translations in Managed Packages

Although you can't edit labels or translations in a managed package, they are controlled by the developer, you can override them with the Translation Workbench. For example, override a custom field in a package.

Translatable Customizations

You can translate only certain components.

Translation Files

Salesforce provides different files types to translate customization labels and review translations.

Considerations for Updating Translated Terms

Translators are responsible for keeping the translated terms updated. When translated components change, the translated value is marked `Out of Date`.

Exporting Translation Files

Create files that contain your org's translatable metadata, such as custom fields, report types, and picklist values. Send these files to your outside translators or translation agency for bulk translation activities, then use **Import** to update your labels.

Import Translated Files

Import and update the translations for your org's metadata, such as custom fields, report types, and picklist values. Typically, the original files are exported from Salesforce, then sent to your outside translators or translation agency for bulk translation activities and returned to you for importing.

Common Errors with Exporting and Importing Translation Files

Troubleshoot issues you might encounter while exporting and importing files in the Translation Workbench.

Best Practices for the Translation Workbench

To make the Translation Workbench most effective, keep these tips in mind.

Enable and Disable the Translation Workbench

The Translation Workbench lets you specify languages you want to translate, assign translators to languages, create translations for customizations you've made to your Salesforce organization, and override labels and translations from managed packages. Everything from custom picklist values to custom fields can be translated so your global users can use all of Salesforce in their language.

1. From Setup, enter *`Translation Settings`* in the `Quick Find` box, then select **Translation Settings**.
2. On the welcome page, click **Enable**.

Note: The "Manage Translation" permission is enabled by default in the System Administrator profile.

To disable the Translation Workbench, from Setup, enter *`Translation Settings`* in the `Quick Find` box, select **Translation Settings**, and then click **Disable**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: In a Developer organization with a managed package containing translations, once the Translation Workbench is enabled, it can't be disabled.

SEE ALSO:

Considerations for Updating Translated Terms

Add Translated Languages and Translators

Add supported languages, activate or deactivate a language, assign translators for that language, and enter translated terms for customizations you've made in your org's default language.

Note: The "Manage Translation" permission is enabled by default in the System Administrator profile.

1. From Setup, enter *Translation Settings* in the Quick Find box, then select **Translation Settings**.
2. Click **Add** to activate a new language or **Edit** to change an existing supported language.
3. If adding a new language, choose a language.
4. To make the entered translations available to your users - select Active. Users can change their personal language anytime whether or not it's active in the Translation Workbench. Selecting Active makes the translations available to the users in that language.

   Tip: We recommend you don't make a language active until the translators have translated all values.

5. To assign translators for this language, select them from the Available List and click **Add**. If you don't see the member you want to add, enter keywords in the search box and click **Find**.

   Important: Ensure all translators have the "View Setup and Configuration" permission so that they can begin translating. Users can only translate languages they're assigned to.

6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To add or edit languages:
- "Manage Translation"

To assign translators:
- "Manage Translation"

SEE ALSO:

Enable and Disable the Translation Workbench

Translate Terms

Considerations for Updating Translated Terms

Translate Terms

1. From Setup, enter *Translate* in the Quick Find box, then select **Translate**.
2. Select the Language you're translating into.
3. Select a Setup Component. Click the pull-down menu to select from the list of translatable customizations. See Translatable Customizations for a complete list of possible customizations.
4. If necessary select an object and aspect. For example, workflow tasks have an object (Account, Contact, etc.) and aspect (Subject or Comment).
5. Double click in the translation column to enter new values. You can press TAB to advance to the next editable field or SHIFT-TAB to go to the previous editable field.

   Note: The Out of Date column indicates that the item has been updated and the term may need translating. When editing a button or link label, you see the Button or Link Name column, which is used to refer to the component when using the SOAP API.

6. Click **Save**.

SEE ALSO:

Considerations for Updating Translated Terms

Add Translated Languages and Translators

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To translate terms:

- "View Setup and Configuration"

  AND

  Be designated as a translator

Override Translations in Managed Packages

Although you can't edit labels or translations in a managed package, they are controlled by the developer, you can override them with the Translation Workbench. For example, override a custom field in a package.

1. From Setup, enter *Override* in the Quick Find box, then select **Override**.
2. Select the Package that you're overriding.
3. Select the Language that you're entering your overrides in.
4. Select a Setup Component. Click the pull-down menu to select from the list of translatable customizations. See Translatable Customizations for a complete list of possible customizations.
5. If necessary select an object and aspect. For example, workflow tasks have an object (Account, Contact, etc.) and aspect (Subject or Comment).
6. Double click in the override column to enter new values. You can press TAB to advance to the next editable field or SHIFT-TAB to go to the previous editable field.

   Note: The Out of Date column indicates that the item has been updated and the term may need to be changed. When editing a button or link label, you see the Button or Link Name column, which is used to refer to the component when using the SOAP API.

7. Double click in the translation column to enter new values. You can press TAB to advance to the next editable field or SHIFT-TAB to go to the previous editable field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To override terms:

- "View Setup and Configuration"

  AND

  "Customize Application"

Note: The `Out of Date` column indicates that the item has been updated and the term may need translating. When editing a button or link label, you see the `Button or Link Name` column, which is used to refer to the component when using the SOAP API.

8. Click **Save**.

SEE ALSO:

Enable and Disable the Translation Workbench

Translatable Customizations

You can translate only certain components.

To view the translatable customizations in your organization, from Setup, enter *`Translate`* in the `Quick Find` box, select **Translate**, and then select a `Setup Component`. Then, if needed, select **Object**, **Custom Report Type Entity**, or **Aspect**.

You can translate the following components.

- Action
- Address Country
- Address State
- Apex Sharing Reason
- Button and Link Label
- Custom App
- Custom Field
- Custom Report Type
- Data Category
- Data Category Group
- Division
- Feed Filter
- Global Picklist
- Layout Section
- Lookup Filter
- Navigation Menu Item (for Communities)
- Lookup Filters
- Picklist Values
- Quick Action Labels
- Record Type
- Related List Label
- Reputation Levels (for Communities)
- Sales Path Step Rich Text
- S-Control
- Solution Category
- Standard Field Help

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Validation Error Message
- Web Tabs
- Workflow Tasks

Important: Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected, and can still be edited.

SEE ALSO:

Enable and Disable the Translation Workbench

Translation Files

Salesforce provides different files types to translate customization labels and review translations.

- Source: Use to translate labels for the first time.
- Untranslated: Use to translate labels after the first translation pass.
- Bilingual: Use to review and edit translations.

Translation files are identified by the extension `.stf`, to represent the Salesforce translation format. A translation filename includes the name of the export option used to create it, the language code for the file's content, and a date stamp.

Multiple `.stf` files created with the Untranslated and Bilingual options are compressed into zip files up to 5 MB in size. If multiple zip files are needed, the zip filenames are each date stamped and incremented. For example, `Untranslated 2010-09-20 05:13 1of2.zip`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Warning: Consider the following when editing your `.stf` files:

- Don't change the `.stf` file extension.
- If you use tabs, new lines, or carriage returns in your text for translation, notice that they are represented with special characters in the `.stf` file format. Tabs are `\t`, new lines are `\n` and carriage returns are `\r`. To ensure consistency between your language versions, ensure these characters are maintained in your translations.

Working with the Source File

Use the Source file to translate an organization's labels for the first time. The Source file contains labels for all of an organization's translatable customizations in the organization's default language.

If you aren't using a standard translation tool such as Trados, work with the file using an application that supports tabs and word wrap, such as WordPad or MS Excel.

Note: If you use MS Excel to enter translations in your `.stf` file, your file format may be corrupted. MS Excel automatically adds quotation marks around entries that have commas. We advise you open your files in a text editor before import and remove these quotation marks if they have been added. The import will fail if these quotation marks are not removed.

To prepare the Source file for your translators:

- Create one copy of the Source file for each language you are translating into.
- In the header of each Source file, change the language code from the organization's default language (such as `en_US`) to the translation language (such as `fr`).

Tell your translators to replace the untranslated values in the LABEL column with translated values.

Note: Don't add columns to or remove columns from the translation file.

Column	Description	Edit Options
KEY	Unique identifiers for labels	Do not edit
LABEL	Labels that are visible to end users	Replace untranslated values with translated values

Working with the Untranslated File

Use the Untranslated file to translate labels that haven't been translated. One Untranslated file is generated for each language. When multiple files are generated, they're exported to a `.zip` file containing `.stf` files for each translation language.

If you aren't using a standard translation tool such as Trados, work with the file using an application that supports tabs and word wrap, such as WordPad or MS Excel.

Note: If you use MS Excel to enter translations in your `.stf` file, your file format may be corrupted. MS Excel automatically adds quotation marks around entries that have commas. We advise you open your files in a text editor before import and remove these quotation marks if they have been added. The import will fail if these quotation marks are not removed.

Tell your translators to replace the untranslated values in the LABEL column with translated values.

Note: Don't add columns to or remove columns from the translation file.

Column	Description	Edit Options
KEY	Unique identifiers for labels	Do not edit
LABEL	Labels that are visible to end users	Replace untranslated values with translated values

Working with the Bilingual File

Use the Bilingual file to review translations, edit labels that have already been translated, and add translations for labels that haven't been translated. One Bilingual file is generated for each translation language.

The TRANSLATED section of the file contains the text that has been translated and needs to be reviewed. The UNTRANSLATED section of the file contains text that hasn't been translated.

Edit the file using an editing application that supports tabs and word wrap, such as WordPad or MS Excel.

Note: If you use MS Excel to enter translations in your `.stf` file, your file format may be corrupted. MS Excel automatically adds quotation marks around entries that have commas. We advise you open your files in a text editor before import and remove these quotation marks if they have been added. The import will fail if these quotation marks are not removed.

- Identify labels that are out of date by scrolling through the OUT OF DATE column to locate values that have an asterisk (`*`). Update out of date labels as needed.
- Edit translated labels in the TRANSLATION column of the TRANSLATED section.
- Replace untranslated labels with translated values in the LABEL column of the UNTRANSLATED section.
- Delete a translation by replacing the desired value in the TRANSLATION column in either section with a left and right angle bracket pair (< >). When the Bilingual file is imported, the label reverts to its original value.

Attention: Don't attempt to delete a translation by deleting a translated label from the file. Deleting a translation in the file doesn't remove the translation from the application after the file is imported.

Note: Don't add columns to or remove columns from the translation file.

Column	Description	Edit Options
KEY	Unique identifiers for labels	Do not edit
LABEL	Labels that are visible to end users	• Do not edit labels in the TRANSLATED section of the file • In the UNTRANSLATED section of the file, replace untranslated labels with translated values
TRANSLATION	Current translation	• In the TRANSLATED section of the file, edit current translations • In the UNTRANSLATED section of the file, add translations
OUT OF DATE	Indicates whether the source text has changed since the previous translation. The out of date indicators are: • An asterisk (*): The label is out of date. A change was made to the default language label and the translation hasn't been updated. • A dash (–): The translation is current.	Do not edit

SEE ALSO:

Exporting Translation Files

Import Translated Files

Considerations for Updating Translated Terms

Translators are responsible for keeping the translated terms updated. When translated components change, the translated value is marked `Out of Date`.

For example, say you enter a translation for an error message on a lookup filter. Later, you edit the lookup filter and change the error message. When you go into the Translation Workbench and view the message translation column, it still shows the translation for the old error message and the checkbox in the `Out of Date` column is selected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

For best results, have your translators check their translations frequently, and be sure to notify them when changes occur. To generate a list of all the translatable customizations and their associated `Out of Date` states, export the translation files.

SEE ALSO:

Translate Terms

Add Translated Languages and Translators

Exporting Translation Files

Create files that contain your org's translatable metadata, such as custom fields, report types, and picklist values. Send these files to your outside translators or translation agency for bulk translation activities, then use **Import** to update your labels.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To export or import translation files

- "Manage Translation"

  AND

  "Customize Application"

  AND

  "Create Documents"

1. From Setup, enter *Export* in the `Quick Find` box, then select **Export**.
2. Select which labels you want to export.
   - **Source**–Used as the initial source for creating new translations.

     Creates a single file that contains a list of all your translatable customizations. Typically, the content is in your organization's default language.

   - **Untranslated**–Used to make updates.

     Creates a set of files that contain only customizations that have not been translated, including new and modified customizations.

     One file is created for each language. These files are then compressed into .zip files.

   - **Bilingual**–Used for reference and reviewing all your untranslated and translated customizations.

     Creates a list of all the translatable labels in their current translated or untranslated state.

     One file is created for each language. These files are then compressed into .zip files.

     The content in each file is divided into Untranslated and Translated sections. Each translatable label is in either the Untranslated or Translated section, according to its translation state. In the Translated section, out of date status for the labels in included.
3. Click **Export**.

   A status message tells you that the export is being processed. When the export is complete an email is sent to the email address specified in your profile.
4. Locate the exported .stf or .zip file.

   Go to ***Your name*** > **Documents** > **Document Folders** > **My Personal Documents** > **Go!**.

   All exported files indicate the Export option used to create them and are date and time stamped. Individual files end with the extension .stf. Multiple files are grouped into .zip files.

   If you have a large number of documents in your personal documents area, you'll find the exported files under the sort letter:

   - B—Bilingual export option, for example: `Bilingual_2010-09-23_11:20.zip`.
   - S—Source export option, for example: `Source_en_US_2010-09-23_11:20.stf`.
   - U—Untranslated export option, for example: `Untranslated_2010-09-23_11:20.zip`.

5. Save the files for translation by your translators or translation agency.

   Click **View** > **Save File** > **OK**. The file is saved to the location specified by your browser. For example, `C:/Users/username/Downloads`.

SEE ALSO:

Import Translated Files

Enable and Disable the Translation Workbench

Import Translated Files

Import and update the translations for your org's metadata, such as custom fields, report types, and picklist values. Typically, the original files are exported from Salesforce, then sent to your outside translators or translation agency for bulk translation activities and returned to you for importing.

Note: Labels that are exported and left as is are not saved as translations on import.

1. Bundle multiple files into .zip files that are no larger than 5 MB. Create multiple .zip files as needed.

   The zipped files don't have to be in the same order or grouping as the exported .zip files.

   For example, you start with two exported .zip files. The first file includes French, Italian, and Japanese. The second file includes Russian, Simplified Chinese, and Greek. You can create:

   - One .zip file with French, Greek, and Italian.
   - One .zip file with Russian and Greek.
   - One .zip file with Simplified Chinese.

2. From Setup, enter *Import* in the `Quick Find` box, then select **Import** under Translation Workbench.
3. Click **Browse** to locate and select the file you want to import.
4. Click **Import**.

   After the import is complete:

   - The labels are updated with the translations.
   - A confirmation email is sent to the email address specified in your profile.

5. Verify the imported changes have been implemented. You can:

   - Check labels in your Salesforce organization.
   - Check labels through the Translation Workbench.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To export or import translation files

- "Manage Translation"

  AND

  "Customize Application"

  AND

  "Create Documents"

SEE ALSO:

Exporting Translation Files

Enable and Disable the Translation Workbench

Common Errors with Exporting and Importing Translation Files

Troubleshoot issues you might encounter while exporting and importing files in the Translation Workbench.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Error Message	What It Means	Troubleshooting Steps
Bilingual File starts with non-section header row: {0}	The header rows of the file you're trying to import are missing, or there is extraneous text (such as notes that aren't commented out) in those rows.	Export your file again and make sure there are header rows for all sections, and that all extraneous text has been commented out or removed from the header rows.
Duplicate key: {0} exists in import file, please re-export.	The specified key appears in your imported file more than once. Each translated item needs to have its own unique key, and each key can only appear in the file one time.	Export your file again and make sure that each key in it is unique, and then re-import the file.
Invalid Key	During translation, Salesforce generates unique keys, or identifiers, for each object, picklist value, or page element you're translating. If these names or keys are changed after you export your file, Salesforce can't match the correct key with the correct name.	Export your file again and make sure the keys in it match those in the file you're trying to import.
Key: {0} could not be uniquely resolved. This is caused by a change to our Custom Report Type Column key format. Please re-export and use the new key format for those keys.	One of the keys in your Custom Report Type (CRT) column is in the wrong format.	Export your file again and make sure you're using the correct CRT key format.
Maximum character limit {x} for {field type} translation exceeded in line:	Each type of field, such as a picklist value, can only have a certain number of characters. Your translated labels for the type of field at the line specified in the error message are too long.	Edit your translated labels so they're within the character limit listed for the field type and import your file again.
No data to import	The file you're trying to import is empty or does not contain any translation changes.	Make sure you're importing the correct file, and that it contains translated data.

Error Message	What It Means	Troubleshooting Steps
No language code specified in file header	The file you're trying to import doesn't have a valid language code, or the language code is in the wrong place.	Make sure your language code is valid and isn't missing or commented out.
No translated or untranslated section header found in the bilingual file	The file you're trying to import is missing section headers.	Make sure your file has section headers and import it again.
No valid file type specified in file header	The file you're trying to import doesn't have a valid import/export type (Source, Untranslated, or Bilingual) specified in the file header. The file type should be in the default language for your organization.	Make sure your file has a valid import/export type in the file header, and that the header did not get translated.
Not a valid file to import. Please select a .stf or a .zip file for import.	You can only import files in .stf format or .zip files that contain .stf files.	Make sure your file is a .stf or a .zip file and try importing it again.
Some keys are appended with their sort order for uniqueness. Re-export your file and ensure that the keys in both files match.	The order of the picklist values in your source file don't match your setup.	Export your source file and match the order of the picklist values to your import file, then import again.
Wrong number of columns in line: {x}. Check that you have escaped tabs (\\t), new lines (\\n), and carriage returns (\\r) in your files.	The file you're importing has extra tabs, new lines, or carriage returns in the line specified in the error message.	Edit your data to remove or escape any extra tabs, newlines, or carriage returns, and make sure the file you're importing has the same number of columns as the file you exported.
Your export request failed. Please retry or contact support.	Salesforce had an unexpected problem while exporting your file.	Contact Salesforce Customer Support.
Your import request failed. Please retry or contact support.	Salesforce had an unexpected problem while importing your file.	Contact Salesforce Customer Support.
Your organization does not have language permissions for {language}.	The file you're trying to import is in a language you haven't yet added to the Translation Workbench.	Add the language you want to use to the Translation Workbench and import your file again.

SEE ALSO:

Exporting Translation Files
Import Translated Files

Best Practices for the Translation Workbench

To make the Translation Workbench most effective, keep these tips in mind.

- Let translators know which languages they are responsible for translating.
- Notify all translators when new translated components are added to your org.
- Advise users when customizing reports or list views to use filter criteria values in their personal language. However, if they use the `starts with` or `contains` operators, advise them to choose the language of the filter criteria values they entered.
- When creating a custom report type that will be translated into multiple languages via the Translation Workbench, we recommend that you set your personal language to match your org's default language. This ensures that words that are translated display in the correct language for translators.
- Salesforce assumes that all customizations are entered in the org's default language. Global administrators should work together in the org's default language.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Enhancing Your Objects, Data, and Fields

Using Custom Object Records

Custom Object Record Overview

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

Custom objects records store information that's unique and important to you and your organization. For example, your organization may use a custom object called "Merchandise" to store data about your company's merchandise. You can also use custom objects for custom applications, such as tracking software enhancements in a development life-cycle.

Your administrator first defines the custom object and its properties, such as custom fields, relationships to other types of data, page layouts, and a custom user interface tab. Once the custom object is created and deployed to users, you can enter data to create individual custom object records. If your administrator has created a tab for the custom object, the tab displays a home page that lets you quickly create and locate custom object records. You can also sort and filter your custom object records using standard and custom list views. In addition, the tab lets you view and edit detailed information on each custom object record to which you have access.

Administrators, and users with the "Modify All Data" permission, can import custom objects.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Custom Objects Home

Custom Objects Home

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

Note: Clicking on a custom object tab displays the tab home page. The name of a custom object tab is the plural form of the custom object as defined by your administrator.

Use the home tab of a custom object to view records.

- To show a filtered list of items, select a predefined list from the `View` drop-down list, or click **Create New View** to define your own custom views. List views let you display a list of records that match specific criteria, such as all custom objects for a tab or just the ones you own. To edit or delete any view you created, select it from the `View` drop-down list and click **Edit**.
- If custom objects are shared with external contacts via Salesforce to Salesforce, choose one of the list views under **[Custom Object Name] from Connections** to view the custom objects that your business partners have shared with you.
- In the **Recent** section, select an item from the drop-down list to display a brief list of the top custom object records matching that criteria. From the list, you can click any custom object name to go directly to the detail. Toggle the **Show 25 items** and **Show 10 items** links to change the number of items that display. The fields you see are determined by the "Custom Object Tab" search layout defined by your administrator and by your field-level security settings (available in Professional, Enterprise, Unlimited, Performance, and Developer Editions). The key list choices are:

Recent Custom Objects Choice	Description
Recently Viewed	The last ten or twenty-five custom object records you viewed, with the most recently viewed item listed first. This list is derived from your recent items and includes records owned by you and other users.
Recently Created	The last ten or twenty-five custom object records you created, with the most recently created item listed first. This list only includes records owned by you.
Recently Modified	The last ten or twenty-five custom object records you updated, with the most recently updated item listed first. This list only includes records owned by you.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view a custom object tab:
- "Read" on the custom object

To view custom object records:
- "Read" on the custom object

SEE ALSO:

Displaying and Editing Custom Object Records

Creating Custom Object Records

Creating Custom Object Records

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

To create a custom object record:

1. Click **New**.
2. Enter the information for the custom object record.
3. Click **Save** when you are finished, or click **Save & New** to save the current record and add another.

If your administrator has defined relationships to other types of records, you can automatically associate the new custom object with another record. View the other record and select the custom object name from the Create New drop-down list in the sidebar, or click **New** *Object Name* in the custom object related list of the other record.

If your organization uses divisions to segment data, custom objects that are detail objects in a master-detail relationship inherit their division from the master object. Custom objects that are not related to other records are automatically in the global division.

SEE ALSO:

Custom Object Record Fields

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view custom object records:

- "Read" on the custom object

To create custom object records:

- "Create" on the custom object

Clone Custom Object Records

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

The **Clone** button on a custom object record creates a custom object record with the same information as the existing record.

Note: When you clone a record, the new record has the record type of the original record. If the record type of the cloned record isn't available in your profile or permission sets, the new record adopts your default record type.

To clone a custom object record:

1. Click **Clone** on an existing custom object record.
2. Enter or change any information for the new record.
3. Click **Save**.

Note: If you have read-only access to a field, the value of that field is not carried over to the cloned record.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To clone custom object records:

- "Create" on the custom object

Deleting Custom Object Records

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

To delete a custom object record, click **Del** next to the record on the custom object's list page or in the custom related list of an associated record. You can also delete a custom object record by clicking **Delete** on the custom object record detail page.

When you delete a custom object record, all related events, tasks, notes, and attachments are also deleted. If the custom object is the master object in a master-detail relationship, any associated detail records are also deleted. If the custom object is the detail side of a master-detail relationship, any associated records are *not* deleted.

The deleted custom object record is moved to the Recycle Bin. If you undelete the record, any related items are also restored.

Note: You can delete a custom object record if you are an administrator, the record owner, or a user above the record owner in the organization role hierarchy, and if you have the appropriate user permission. Custom objects that are on the detail side of a master-detail relationship do not have an `Owner` field and can be deleted by any user who has access to edit the associated master record.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To delete custom object records:

- "Delete" on the custom object

SEE ALSO:

Object Relationships Overview

Displaying and Editing Custom Object Records

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

Note: Clicking on a custom object tab displays the tab home page. The name of a custom object tab is the plural form of the custom object as defined by your administrator.

Displaying Custom Object Records

Once you have located a custom object record on the custom objects tab home or list pages, click the custom object record name to display detailed information.

If the Salesforce console is set up to include custom objects, select it from the Force.com app menu to find, view, and edit custom object records and their related items on one screen.

Editing Custom Object Records

To update custom object record details, click **Edit**, and then change the fields you want to update. When you have finished, click **Save**. You can also click **Save & New** to save the current custom object record and create another. Required fields are marked with red.

Note: You cannot change the `Object Name` or `Data Type` if the custom object is referenced in Apex.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view custom object records:

- "Read" on the custom object

To change custom object records:

- "Edit" on the custom object

Custom Object Related Lists

The lower portion of the custom object record detail page provides information related to the custom object record, including activities, notes, attachments, and any other related records. The related lists you see are determined by your personal customization, and by any customization your administrator has made to page layouts or your permissions to view related data. You can click on individual items to display additional detail. Click **more** at the bottom of the page or **View More** below a related list to display more items. Clicking **New** lets you directly add new items.

Tip: To help your users see record details and related lists more easily, enable Hover Details in the User Interface panel of Setup.

Printing Custom Object Records

On a custom object detail page, click **Printable View** to print the record.

To return to the last list page you viewed, click **Back to list** at the top of any custom object record detail page. If your organization has enabled collapsible page sections, use the arrow icons next to the section headings to expand or collapse each section on the detail page.

SEE ALSO:

Creating Custom Object Records

Sharing Custom Object Records

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

The administrator defines the default sharing model for an entire organization. You can change this model to extend sharing to more users than the default set. However, you cannot change the sharing model to make it more restrictive than the default.

To view and manage sharing details, click **Sharing** on the custom object record detail page. The Sharing Detail page lists the users, groups, roles, and territories that have sharing access to the record. On this page, you can:

- Grant access to the record to other users, groups, roles, or territories

  Note: You can't share a custom object record with another user unless that user has the "Read" permission on the custom object.

- View all users who have access to the record
- Edit or delete the record's access level

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view a custom object record:

- "Read" on the custom object

Custom Object Record Fields

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

Custom object record fields contain the data you enter for your custom object records. Here is a description of the fields (in alphabetical order) that make up a custom object record. Some of these fields may not be visible or editable depending on your page layout and field-level security settings.

Your administrator may also have defined additional custom fields.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Created By	User who created the record.
Currency	Currency of the record if multicurrency is enabled.
Division	Division to which the custom object record belongs. Custom objects that are "detail" objects in a master-detail relationship inherit their division from the master object. Custom objects that are not related to other records are automatically in the global division. Available only in organizations that use divisions to segment their data.
Last Modified By	User who most recently changed the record.
Name	Identifier for the custom object record. This name appears in page layouts, related lists, lookup dialogs, search results, and key lists on tab home pages. By default, this field is added to the custom object page layout as a required field.
Owner	Assigned owner of the custom object record. If the custom object becomes the detail side of a master-detail relationship, this field is removed, as ownership of the data is controlled by the master object, or by the primary master object for a custom object with two master-detail relationships. Note: Custom objects on the "detail" side of a master-detail relationship can't have sharing rules, manual sharing, or

Field	Description
	queues, as these require the `Owner` field.

SEE ALSO:

Custom Objects Home

Using Custom Related Lists

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

Custom related lists display on the lower portion of the detail page for another record. They list the custom object records that are associated with that record.

From a custom related list, you can:

- Click the custom object record name to view detailed information.
- Click **Edit** or **Del** to edit or delete the custom object record.
- Click **New** to create a new custom object record that is associated with the record you are viewing.

SEE ALSO:

Custom Object Record Overview

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view custom object record related lists:

- "Read" on the custom object

  AND

  "Read" on the associated record

To change custom object records:

- "Edit" on the custom object

To create custom object records:

- "Create" on the custom object

To delete custom object records:

- "Delete" on the custom object

Viewing Custom Object Lists

Important: You might be viewing this page because your Salesforce administrator didn't create custom help. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects. The Salesforce Help covers only the standard objects provided with the initial Salesforce integration.

The custom objects list page displays a list of custom object records in your current view. A list view includes custom object records of one type only; you cannot view more than one type of custom object in a single list view.

- Click a custom object name to view the custom object record detail. Click **Edit** or **Del** next to a custom object record name to edit or delete the record.
- To take ownership of custom object records in a queue, view the queue list view, check the box next to one or more records, and then click **Accept**.

  Note: You can view and accept records only from queues you belong to, or if you are higher in the role hierarchy than a queue member. Administrators, users with the "Modify All" object-level permission for the given object, and users with the "Modify All Data" permission, can view and take records from any queue.

- To transfer ownership of multiple records at once, select the box next to one or more custom object records and click **Change Owner**. Enter the new record owner, optionally select `Send Notification Email` to send an email to the new record owner, and click **Save**.
- Click **New** *`Custom Object Name`* or select the custom object name from the Create New drop-down list in the sidebar to create a custom object of that type.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view custom object lists:
- "Read" on the custom object

To create custom object records:
- "Create" on the custom object

SEE ALSO:

Custom Object Record Overview

Administering Custom Objects

Custom Objects

Store information that's unique to your organization. Choose whether your custom objects are searchable, support sharing, or include access to the Bulk API and Streaming API.

Every custom object is classified as either an *Enterprise Application object* or a *Light Application object*. The difference between these two categories is that Light Application objects don't support sharing, access to the Bulk API, or access to the Streaming API.

If you need to track your organization's usage of each category, create a custom report type with a primary object of User Licenses and a child object of Custom Object Usage by User License Metrics.

By default, all custom objects are Enterprise Application objects. To make your custom object a Light Application object, disable `Allow Sharing`, `Allow Bulk API Access`, and `Allow Streaming API Access` on the object's detail page.

After you define a custom object, you can:

- Create custom fields.
- Associate the custom object with other records and display the custom object data in custom related lists.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Track tasks and events for custom object records.
- Build page layouts.
- Customize the search results and the custom object fields that display in them.
- Create a custom tab for the custom object.

  Note: To allow your users to find a custom object's records when they search, create a custom tab set to `Default On` or `Default Off`. Creating a custom tab enables the custom object's **Allow Search** setting.

- Create reports and dashboards to analyze custom object data.
- Share your custom objects, custom tabs, custom apps, and any other related components with other Salesforce users on Force.com AppExchange.
- Import custom object records.

SEE ALSO:

Tip sheet: Building Custom Objects, Tabs, and Related Lists

Track Your Organization's Custom Object Usage by User License Type

Manage Custom Objects

Create, customize, edit, delete, or truncate custom objects to extend the functionality that standard objects, like accounts and contacts, provide.

Note: Your administrator may have created a tab without any help. If you need help to understand how a tab for a custom object works, contact your administrator.

Your object management settings list the custom objects that are defined for your organization. From this list, you can:

- Define a custom object.
- Display detailed information about a custom object.

  Optional features you can customize include enabling search and reports, tracking activities, tracking field history, and making the object available for the Salesforce Customer Portal.

- To update the custom object definition, click **Edit** and update the desired fields.

  Note: The `Allow Reports`, `Allow Activities`, and `Allow Search` fields are not locked in Managed - Released and can be changed by the developer in future releases of a managed package.

- To delete a custom object, click **Del**.
- To truncate a custom object, click **Truncate**.
- To view deleted custom objects, click the **Deleted Objects** link. The total number of deleted custom objects for your organization is listed in parentheses.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Managed Packages are not available in **Database.com**.

USER PERMISSIONS

To create and edit custom objects:

- "Customize Application"

The detail page of the custom object provides information about various characteristics of the object, including standard fields, custom fields, field history tracking, relationships, custom links, search layouts, page layouts, and object limits. You can:

- Click individual items to display additional detail.
- To delete a custom field, click **Del** next to its name in the Custom Fields & Relationships section.
- Click **More** at the bottom of the page or **View More** below a related list to display more items.

- Click **New** to directly add new items.

Note: The object limit percentages are truncated, not rounded. For example, if your org uses 95.55% of the limit for a particular customization, the object limit displays 95%.

SEE ALSO:

Custom Objects

Modify Custom Objects

Tip sheet: Building Custom Objects, Tabs, and Related Lists

Find Object Management Settings

Deployment Status for Custom Objects and External Objects

While developing a custom object or external object, you might not want users to see and interact with it. Because users might get frustrated with changes in layout or lose data when you delete custom fields, control visibility of the new object until you are finished.

Use the `Deployment Status` setting in the object definition to control when users can see and use the object and its associated custom tab, related lists, and reports.

- Choose "In Development" as the `Deployment Status` when first creating your custom object or external object. Doing so hides it from users while you are designing and testing it. Only users with the "Customize Application" permission can see the object tab, search results, related lists, and report data types.
- Change the `Deployment Status` to "Deployed" when you want to allow all users to use the object and any associated custom tab, related lists, and reports.
- If you make more enhancements after deploying a custom object or external object, you can change the `Deployment Status` back to "In Development."

Note: A custom report type's `Deployment Status` changes from `Deployed` to `In Development` if its primary object is a custom or external object whose `Deployment Status` similarly changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To deploy custom objects and external objects:

- "Customize Application"

Define a Custom Object

Track and store data that's unique to your organization. Follow different steps, depending on which Salesforce experience you're using.

IN THIS SECTION:

Define a Custom Object in Lightning Experience

Track and store data that's unique to your organization. If you see the App Launcher icon ( ) on the left side of the navigation bar at the top of your screen, you're in Lightning Experience. If not, you're in Salesforce Classic.

Define a Custom Object in Salesforce Classic

Track and store data that's unique to your org.

Fields for Defining Custom Objects

When you create a custom object, several fields are required to define how you can access the object.

Considerations for Creating Custom Objects

Before you create a custom object, make sure that you've reviewed the considerations.

SEE ALSO:

Object Relationships Overview

Define Object-Level Help

Define a Custom Object in Lightning Experience

Track and store data that's unique to your organization. If you see the App Launcher icon (⁞⁞⁞) on the left side of the navigation bar at the top of your screen, you're in Lightning Experience. If not, you're in Salesforce Classic.

1. From the top right corner of any page in Setup, click **Create** > **Custom Object**.
2. Complete the fields for your custom object.
3. Save the new object.

SEE ALSO:

Fields for Defining Custom Objects

Considerations for Creating Custom Objects

Define a Custom Object

EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and edit custom objects:

- "Customize Application"

Define a Custom Object in Salesforce Classic

Track and store data that's unique to your org.

1. From Setup, enter *Objects* in the `Quick Find` box, then select **Objects**.
2. Click **New Custom Object**.
3. Follow the wizard to complete the fields for your custom object.
4. Save the new object.

SEE ALSO:

Fields for Defining Custom Objects

Considerations for Creating Custom Objects

Define a Custom Object

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and edit custom objects:

- "Customize Application"

Fields for Defining Custom Objects

When you create a custom object, several fields are required to define how you can access the object.

Note: If an administrator created a tab without including help, contact your administrator if you need help with how a custom object works.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Label`	This name is used to refer to the object in a user interface page.
`Plural Label`	The plural name of the object. If you create a tab for this object, this name is used for the tab.
`Gender`	If it is appropriate for your organization's default language, specify the gender of the label. This field appears if the organization-wide default language expects gender. Your personal language preference setting does not affect whether the field appears. For example, if the organization's default language is English and your personal language is French, you aren't prompted for gender when creating a custom object.
`Starts with a vowel sound`	If it is appropriate for your organization's default language, indicate whether the label is preceded by "an" instead of "a."
`Object Name`	A unique name used to refer to the object when using the API. In managed packages, this name prevents naming conflicts with package installations. Use only alphanumeric characters and underscores. The name must begin with a letter and have no spaces. It cannot end with an underscore nor have two consecutive underscores.
`Description`	An optional description of the object. A meaningful description helps you remember the differences between your custom objects when you are viewing them in a list.
`Context-Sensitive Help Setting`	Defines what displays when a user clicks **Help for this Page** from the custom object record home (overview), edit, and detail pages, list views, and related lists. To display the standard Salesforce help available for any custom object record, select **Open the standard Salesforce Help & Training window**.

Field	Description
	To display custom object-level help for your custom object, select **Open a window using a Visualforce page** and then select the Visualforce page to use as the target of the context-sensitive help link from that custom object's pages. Note: This setting doesn't affect the Help & Training link at the top of a page. That link always opens the Salesforce Help & Training window.
`Record Name`	The name used in page layouts, list views, related lists, and search results.
`Data Type`	The type of field (text or auto-number) for the record name. Records that have unique IDs instead of names are auto-numbered and always a read-only field.
`Display Format`	For an auto-numbered record name, enter the display format. You can have up to two sets of curly braces. For more information, see Custom Field Attributes on page 134.
`Starting Number`	For an auto-numbered record name, enter the number to use when creating your first record for this custom object.
`Allow Reports`	Makes the data in the custom object records available for reporting purposes. To create reports on custom objects, choose the **Other Reports** report type category, unless the custom object has a relationship with a standard object. When the custom object has a master-detail relationship with a standard object or is a lookup object on a standard object, select the standard object for the report type category instead.
`Allow Activities`	Allows users to associate tasks and scheduled calendar events related to the custom object records.
`Allow in Chatter Groups`	Allows users to add records of this custom object type to Chatter groups. When `true`, users with permissions can create records of this object type using the group publisher. The created record is associated with the group and appears in the group record list. When `false`, users with permissions can use the group publisher to create records of this object type, but the record is not associated with the group.
`Enable Divisions`	If your organization has divisions enabled, select this option to enable the custom object for divisions. Divisions group records for simplified search results, list views, reports, and other areas within Salesforce. Salesforce adds a `Division` field to the custom object. If the custom object is the master in a master-detail

Field	Description
	relationship, custom objects on the detail side also get the `Division` field and inherit their division from the master record.
`Available for Customer Portal`	Makes the custom object available to all portal users. This option is available only if your organization has a customer portal. If you enable Communities in your organization, this option no longer appears, and all custom objects are available in your communities. If, before enabling Communities, you had a Customer Portal and custom objects without this option selected, those objects become available in your Customer Portal.
`Track Field History`	Enables your organization to track changes to fields on the custom object records. For example, it tracks who changed the field value and when, what the value was before the edit, and what it was changed to. History data is available for reporting, so users can easily create audit trail reports when this feature is enabled.
`Allow Sharing`	When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object. When this setting is enabled, you must also enable `Allow Bulk API Access` and `Allow Streaming API Access`.
`Allow Bulk API Access`	When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object. When this setting is enabled, you must also enable `Allow Sharing` and `Allow Streaming API Access`.
`Allow Streaming API Access`	When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object. When this setting is enabled, you must also enable `Allow Bulk API Access` and `Allow Sharing`.
`Deployment Status`	Indicates whether the custom object is visible to other users.
`Allow Search`	To allow your users to find a custom object's records when they search, create a custom tab set to `Default On` or `Default Off`. Creating a custom tab enables the custom object's **Allow Search** setting. Custom object records are searchable in the Salesforce user interface only if the custom object is associated with a custom tab. Users aren't required to add the tab for display.

Field	Description
`Add Notes & Attachments...`	Allows users to attach notes and attachments to custom object records. You can attach external documents to any object record in much the same way that you can add a PDF file or photo as an attachment to an email. This option is available only when you are creating a new object.
`Launch the New Custom Tab Wizard`	Starts the custom tab wizard after you save the custom object.

SEE ALSO:

Define a Custom Object

Considerations for Creating Custom Objects

Before you create a custom object, make sure that you've reviewed the considerations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Object Permissions

In Enterprise, Unlimited, Performance, Professional, and Developer Editions, when you create a custom object, the "Read," "Create," "Edit," and "Delete" permissions for that object are disabled for profiles that have "View All Data" or "Modify All Data" disabled. Enable access to custom objects in permission sets or custom profiles, and assign them to the users who need access.

In Contact Manager and Group Editions, when you create a custom object, the "Read," "Create," "Edit," and "Delete" permissions for that object are enabled for all profiles.

Sharing Model

An org-wide default setting controls the data sharing model for custom objects. For more information, see Custom Object Security on page 107.

Delegating Custom Object Administration

After you create a custom object, you can delegate its administration to non-admin users.

Queues

After you create a custom object, you can define queues to distribute ownership of custom object records to your users.

SEE ALSO:

Define a Custom Object

Modify Custom Objects

Customize the user interface for your custom objects by:

- Creating a custom tab (see Custom Tabs)
- Creating custom fields and relationships (see Define Custom Object Fields on page 89)
- Adding customized buttons and links to perform actions or link to other pages or websites (see Define Custom Buttons and Links on page 906)
- Defining which fields display for users on record detail and edit pages (see Build Page Layouts for Custom Objects on page 28)
- Specifying which fields display for users in search results, lookup dialogs, and in the key lists on custom object tabs (see Customize Search Layouts for Custom Objects on page 107)
- Creating record types to display different picklist values and page layouts to different users based on their profiles (see Create Record Types for Custom Objects on page 106)

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To customize custom objects:

- "Customize Application"

Define Custom Object Fields

Custom object fields store the data for your custom object records.

Custom Fields for Custom Objects

You can create custom fields to store information unique to your organization. You can also create custom relationship fields to associate your custom object with another object in Salesforce.

Standard Fields for Custom Objects

Custom objects automatically include the following standard fields. Click **Edit** to modify any of the editable fields.

Field	Description
`Created By`	User who created the record.
`Currency`	Currency of the record if multicurrency is enabled.
`Division`	Division to which the custom object record belongs. Custom objects that are "detail" objects in a master-detail relationship inherit their division from the master object. Custom objects that are not related to other records are automatically in the global division. Available only in organizations that use divisions to segment their data.
`Last Modified By`	User who most recently changed the record.
`Name`	Identifier for the custom object record. This name appears in page layouts, related lists,

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Divisions are not available in **Database.com**.

USER PERMISSIONS

To view and edit standard fields:

- "Customize Application"

To create custom fields:

- "Customize Application"

Field	Description
	lookup dialogs, search results, and key lists on tab home pages. By default, this field is added to the custom object page layout as a required field.
`Owner`	Assigned owner of the custom object record. If the custom object becomes the detail side of a master-detail relationship, this field is removed, as ownership of the data is controlled by the master object, or by the primary master object for a custom object with two master-detail relationships. Note: Custom objects on the "detail" side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the `Owner` field.

Delete Custom Objects

When you delete a custom object, Salesforce does not add it to the Recycle Bin. Instead, deleted objects appear in the Deleted Objects list for 15 days. During this time, the object and its data are *soft deleted*, meaning you can restore or permanently erase (*hard delete*) the object and its data. After 15 days, the object and its data are automatically hard deleted.

Soft-deleted custom objects and their data count against your organization's limits; hard-deleted items do not.

To delete a custom object:

1. From the object management settings for custom objects, click **Del** next to object that you want to delete.
2. When prompted, select the `Yes, I want to delete the custom object` checkbox to confirm and click **Delete**.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To delete custom objects:
- "Customize Application"

Why Can't I Delete a Certain Custom Object?

You can't delete a custom object if it:

- Is on the master side of a master-detail relationship.
- Contains custom fields that are used in a roll-up summary field on another object.
- Is referenced in Apex, a Visualforce page, or an reporting snapshot.
- Is referenced by a duplicate rule or a matching rule.
- Contains more than 100,000 records. If the object you want to delete has more than 100,000 records, first delete an appropriate number of records and then delete the object.

Results of Deleting Custom Objects

When you delete a custom object, Salesforce:

- Displays an `Insufficient Privileges` message if someone clicks a bookmark to a deleted custom object record's URL.
- Removes the object from Force.com AppExchange packages.
- Changes the master-detail relationship to a lookup relationship, if the deleted object is on the detail side of a master-detail relationship.

- Removes or erases:
  - The object's custom tab
  - List views and workflow rules for the object
  - Mobile configuration settings, including data sets, mobile views, and excluded fields
  - Standard report types associated with the object, and reports based on standard report types if the deleted object is on the detail side of a master-detail relationship
- Hides, inactivates, or disables:
  - The custom object definition and related definitions
  - The object's records and related records, including any records in the Recycle Bin
  - Custom report types for which the deleted object is the main object
  - Custom reports for which the deleted object is the main object
  - Custom formula fields on the object
  - Custom validation rules and approval processes on the object

Note: Many removed, hidden, inactive, or disabled items can be restored if you undelete the custom object. See Manage Deleted Custom Objects for information about restoring deleted custom objects.

Results of Hard Deleting Custom Objects

When a custom object is hard deleted, either manually, or automatically after 15 days:

- The custom object's definition and data are permanently deleted, and can't be restored.
- The custom object and its data no longer count against your organization's limits.
- If the deleted object is on the detail side of a master-detail relationship, master records currently in the Recycle Bin aren't restorable if one or more detail records were automatically deleted as a result of the master record being deleted. Attempting to undelete the master record results in an error.

  Note: This only happens when the deleted detail records have their custom object definition hard deleted while the master record is in the Recycle Bin.

SEE ALSO:

Manage Deleted Custom Objects

Find Object Management Settings

Manage Deleted Custom Objects

Deleted custom objects appear in the Deleted Objects list for 15 days. During this time, you can choose to permanently delete the object and its data, or you can undelete them. If you undelete a custom object, you might need to do some manual cleanup to restore list views and other customizations that use the object.

- To view a list of deleted custom objects:
  1. Go to the object management settings for custom objects.
  2. Click **Deleted Objects** at the bottom of the list.

  The Deleted Objects link appears only when you have at least one deleted custom object in your organization. The number in parentheses indicates the total number of deleted custom objects.
- In the Deleted Objects list, you can do any of the following:
  - Click the object's label to view details about it
  - Click **Erase** to permanently remove the object and its data
  - Click **Undelete** to restore the object and its data

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To restore deleted custom objects:
- "Customize Application"

To permanently delete custom objects:
- "Customize Application"

Results of Hard Deleting Custom Objects

When a custom object is hard deleted, either manually, or automatically after 15 days:

- The custom object's definition and data are permanently deleted, and can't be restored.
- The custom object and its data no longer count against your organization's limits.
- If the deleted object is on the detail side of a master-detail relationship, master records currently in the Recycle Bin aren't restorable if one or more detail records were automatically deleted as a result of the master record being deleted. Attempting to undelete the master record results in an error.

  Note: This only happens when the deleted detail records have their custom object definition hard deleted while the master record is in the Recycle Bin.

Limitations for Restoring Truncated Custom Objects

Copies of truncated custom objects also appear in the list of deleted objects. Truncated custom objects can't be restored to their original state. Undeleted copies of truncated objects have a new name and new URL, and some fields and data *cannot* be manually restored.

Restoring a Custom Object to Its Predeleted State

When you restore a deleted custom object, its records are also undeleted, including any that were in the Recycle Bin.

Note: It might take several hours before you can search the undeleted object's records.

To ensure that you return the object to its fully functional, predeleted state, check all affected conditions and customizations, and manually fix them if necessary.

AppExchange packages
Add the custom object to any appropriate Force.com AppExchange packages.

Custom tabs
Re-create a custom tab for the object and add it to any custom apps that use it.

List views, reports, and workflow rules
Re-create them.

Validation rules and approval processes
Re-activate them.

Formula fields
Open and save any custom formula fields on the object to re-enable them.

Page layouts
Page layouts are restored automatically on the undeleted object. Page layouts are also restored automatically on other objects that use the page layout in a related list—as long as the page layout wasn't edited while the object was deleted. Otherwise, you have to add the related list back to the other object.

Custom report types
For custom report types where the deleted object was not the main object, add the reference back to the restored object. Reports based on the custom report type are automatically restored if they weren't edited while the object was deleted. Re-create any reports that have been edited.

Relationships
If the deleted custom object was on the detail side of a master-detail relationship, Salesforce converted it to a lookup relationship. Change it back to master-detail.

Developer name
The developer name for the object was changed to *objectname*_del. Change it back to the original name, *objectname*_c, so that customizations using the name will work properly.

Deployment status
When the custom object was deleted, its `Deployment Status` field was set to In Development. When you've restored all affected customizations to the undeleted object, change its status back to Deployed.

SEE ALSO:

Delete Custom Objects

Truncating Custom Objects

Find Object Management Settings

Truncating Custom Objects

It's important to understand what truncating an object does before using it to remove records.

Truncating a custom object lets you remove all of the object's records, while keeping the object and its metadata. Truncating custom objects is similar to the mass delete option available for standard objects.

When you truncate a custom object, all of the object's records are removed permanently, but the object's definition remains. The records no longer count against your org limits.

In contrast, if you *delete* a custom object, the object moves to the Deleted Objects list for 15 days. After that the object and its records are permanently deleted.

Important: Truncated custom objects can't be restored to their original state.

You can't truncate standard objects or custom objects that:

- Are referenced by another object through a lookup field or that are on the master side of a master-detail relationship

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

- Are referenced in a reporting snapshot
- Have a custom index or an external ID
- Have activated skinny tables

In addition, you can't truncate custom objects when your org has reached its limit on allowed custom objects.

Truncating a custom object erases:

- All records currently sitting in the custom object's Recycle Bin
- The custom object's history
- Related events, tasks, notes, and attachments for each deleted record

Truncating a custom object breaks:

- Bookmarks to the truncated objects and its records. If someone clicks a bookmark to the truncated custom object or to a deleted record's URL, Salesforce displays an `Insufficient Privileges` message.
- Apex scripts and Visualforce pages with references to a truncated object or record.

After truncating a custom object, you can continue to use the custom object and add new records. Salesforce preserves:

- The custom object definition and all related definitions
- Workflow rules, actions, and triggers
- Sharing rules associated with the custom object
- Validation rules and approval processes
- Master-detail relationships and formula fields
- Translations
- Mobile configuration settings

When working with truncated objects, keep in mind:

- The truncated object tab has a new URL, so new bookmarks need to be created.
- List views and reports need to be refreshed after truncation.
- Roll-up summary fields need to be recalculated after truncation.
- There is no support for truncation in the API.
- To truncate objects that contain master-detail relationships, first truncate the detail (child) objects and then the (master) parent objects, working your way up the relationship tree.

SEE ALSO:

Truncate Custom Objects

Manage Deleted Custom Objects

Truncate Custom Objects

Truncating custom objects allows you to permanently delete all of the object's records but preserve the empty object and its metadata.

Important: Truncating custom objects causes some irreversible changes to the truncated object and its records. Before truncating, see Truncating Custom Objects. Then, enable it for your organization by entering *User Interface* in the Quick Find box, selecting **User Interface**, and then selecting the permission.

Truncating custom objects is a fast way to permanently remove all of the records from a custom object, while keeping the object and its metadata intact for future use. Truncating is useful, for example, if you have created a custom object and filled it with test records. When you're done with the test data, you can truncate the object to purge the test records, but keep the object and put it into production. This is much faster than batch-deleting records and possibly recreating the object.

1. Go to the object management settings for custom objects.
2. Click an object name to go to the object's detail page, and then click **Truncate**.
3. In the Confirm Custom Object Truncate window, review the warning and then enter the name of the object to truncate in the empty field.
4. Click **Truncate**.

SEE ALSO:

Manage Deleted Custom Objects

Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To truncate custom objects:

- "Customize Application"

Track Your Organization's Custom Object Usage by User License Type

To keep track of how many custom objects your users are assigned to, create a custom report type with the User Licenses and Custom Object Usage by User License Metrics objects.

1. From Setup, enter *Report Types* in the Quick Find box, then select **Report Types**.
2. Click **New Custom Report Type.**
3. Fill out the fields.

For this field...	Enter or select...
Primary Object	User Licenses
Report Type Label	A label for this report type. The label appears when users create a report. For example, *Custom Object Usage by User License*.
Report Type Name	A unique name for this report type.
Description	A description for this report type. This description appears when users create a report.
Store in Category	Other Reports

4. (Optional) To make this report type available to other users, select a Deployment Status of Deployed.
5. Click **Next**.
6. Click the box under the primary object.
7. Select Custom Object Usage by User License Metrics.
8. Click **Save**.
9. Create a report using the new report type.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom objects are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Custom report types are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or update custom report types:
- "Manage Custom Report Types"

To delete custom report types:
- "Modify All Data"

Object Relationships Overview

Create relationships to link objects with each other, so that when your users view records, they can also see related data. For example, link a custom object called "Bugs" to cases to track product defects that are associated with customer cases.

You can define different types of relationships by creating custom relationship fields on an object. Before you begin creating relationships, determine the type of relationship that suits your needs.

Different types of relationships between objects in Salesforce determine how they handle data deletion, sharing, and required fields in page layouts. Let's review the types of relationships.

Master-detail
Closely links objects together such that the master record controls certain behaviors of the detail and subdetail record. For example, you can define a two-object master-detail relationship,

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

such as Account—Expense Report, that extends the relationship to subdetail records, such as Account—Expense Report—Expense Line Item. You can then perform operations across the master—detail—subdetail relationship.

Behaviors of master-detail relationships:

- Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it, and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later, separately, delete its master record, you cannot undelete the detail record, as it no longer has a master record to relate to.
- By default, records can't be reparented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the `Allow reparenting` option in the master-detail relationship definition.
- The `Owner` field on the detail and subdetail records is not available and is automatically set to the owner of the master record. Custom objects on the "detail" side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the `Owner` field.
- Detail and subdetail records inherit security settings and permissions from the master record. You can't set permissions on the detail record independently.
- The master-detail relationship field (which is the field linking the objects) is required on the page layout of the detail and subdetail records.
- The master object can be a standard object, such as Account or Opportunity, or a custom object.
- As a best practice, don't exceed 10,000 child records for a master-detail relationship.

Many-to-many

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you may have a custom object called "Bug" that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

Lookup

Links two objects together. Lookup relationships are similar to master-detail relationships, except they do not support sharing or roll-up summary fields. With a lookup relationship, you can:

- Link two different objects.
- Link an object with itself (with the exception of the user object; see Hierarchical on page 98). For example, you might want to link a custom object called "Bug" with itself to show how two different bugs are related to the same problem.

Note: Lookup relationships from objects related to the campaign member object aren't supported; however, you can create lookup relationships from the campaign member object related to other objects.

When you create a lookup relationship, you can set these options:

- Make the lookup field required for saving a record, requiring it on the corresponding page layout as well.
- If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:
  - `Clear the value of this field` This is the default. Clearing the field is a good choice when the field does not have to contain a value from the associated lookup record.
  - `Don't allow deletion of the lookup record that's part of a lookup relationship` This option restricts the lookup record from being deleted if you have any dependencies, such as a workflow rule, built on the relationship.
  - `Delete this record also` Available only if a custom object contains the lookup relationship, not if it's contained by a standard object. However, the lookup object can be either standard or custom. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data. For example, say you have an expense

report record with a lookup relationship to individual expense records. When you delete the report, you probably want to delete all of the expense records, too.

Warning: Choosing `Delete this record also` can result in a *cascade-delete*. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted *even if they don't have access to the records*. To prevent records from being accidentally deleted, cascade-delete is disabled by default. Contact Salesforce to get the cascade-delete option enabled for your organization.

Cascade-delete and its related options are not available for lookup relationships to business hours, community, lead, price book, product, or user objects.

When you define a lookup relationship, you have the option to include a lookup field on the page layouts for that object as well as create a related list on the associated object's page layouts. For example, if you have a custom object called "PTO Requests" and you want your users to link a PTO request with the employee submitting the request, create a lookup relationship from the PTO Request custom object with the user object.

If the parent record in a lookup relationship is deleted, the field history tracking for the child record does not record the deletion. For example, if a parent account is deleted, the Account History related list for the child account does not show the deletion.

You can't delete an object or record in a lookup relationship if the combined number of records between the two linked objects is more than 100,000. To delete an object or record in a lookup relationship, first delete an appropriate number of its child records.

External lookup

An external lookup relationship links a child standard, custom, or external object to a parent external object. When you create an external lookup relationship field, the standard External ID field on the parent external object is matched against the values of the child's external lookup relationship field. External object field values come from an external data source.

Indirect lookup

An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values come from an external data source.

Hierarchical

A special lookup relationship available for only the user object. It lets users use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.

Tip: When creating a hierarchical field in Personal, Contact Manager, Group, and Professional Editions, you can select the `Restricted Field` checkbox so that only users with the "Manage Internal Users" permission can edit it. In Professional, Enterprise, Unlimited, Performance, and Developer Edition, use field-level security instead.

SEE ALSO:

Considerations for Relationships

External Object Relationships

Create a Many-to-Many Relationship

Define a Custom Object

Create a Many-to-Many Relationship

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you may have a custom object called "Bug" that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs. When modeling a many-to-many relationship, you use a *junction object* to connect the two objects you want to relate to each other.

Junction Object
A custom object with two master-detail relationships. Using a custom junction object, you can model a "many-to-many" relationship between two objects. For example, you may have a custom object called "Bug" that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

Creating the many-to-many relationship consists of:

1. Creating the junction object.
2. Creating the two master-detail relationships.
3. Customizing the related lists on the page layouts of the two master objects.
4. Customizing reports to maximize the effectiveness of the many-to-many relationship.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Reports are not available in **Database.com**.

USER PERMISSIONS

To create a many-to-many relationship:
- "Customize Application"

Creating the Junction Object

1. Create a custom object to be your junction object.
2. In the custom object wizard, consider these tips specifically for junction objects:
   - Name the object with a label that indicates its purpose, such as `BugCaseAssociation`.
   - For the `Record Name` field, it is recommended that you use the auto-number data type.
   - Do not launch the custom tab wizard before clicking **Save**. Junction objects do not need a tab.

Creating the Two Master-Detail Relationships

To create the two master-detail relationships:

1. Verify that the two objects you want to relate to each other already exist. For example, you may want to relate the standard case object to a custom bug object.
2. On the junction object, create the first master-detail relationship field. In the custom field wizard:
   a. Choose `Master-Detail Relationship` as the field type.
   b. Select one of the objects to relate to your junction object. For example, select `Case`.

      The first master-detail relationship you create on your junction object becomes the *primary* relationship. This affects the following for the junction object records:

      - Look and feel: The junction object's detail and edit pages use the color and any associated icon of the primary master object.
      - Record ownership: The junction object records inherit the value of the `Owner` field from their associated primary master record. Because objects on the detail side of a relationship do not have a visible `Owner` field, this is only relevant if you later delete both master-detail relationships on your junction object.
      - Division: If your organization uses divisions to segment data, the junction object records inherit their division from their associated primary master record. Similar to the record ownership, this is only relevant if you later delete both master-detail relationships.

   c. Select a `Sharing Setting` option. For master-detail relationship fields, the `Sharing Setting` attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.

   d. For the `Related List Label` that will display on the page layout of the master object, do not accept the default. Change this to use the name of the other master object in your many-to-many relationship. For example, change this to *`Bugs`* so users will see a Bugs related list on the case detail page.

3. On the junction object, create the second master-detail relationship. In the custom field wizard:

   a. Choose `Master-Detail Relationship` as the field type.

   b. Select the other desired master object to relate to your junction object. For example, select *`Bug`*.

      The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

   c. Select a `Sharing Setting` option. For master-detail relationship fields, the `Sharing Setting` attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.

   d. For the `Related List Label` that will display on the page layout of the master object, do not accept the default. Change this to use the name of the other master object in your many-to-many relationship. For example, change this to *`Cases`* so users will see a Cases related list on the bug detail page.

Customizing Many-to-Many Relationship Related Lists

For a many-to-many relationship in Salesforce, each master object record displays a related list of the associated junction object records. To create a seamless user experience, you can change the name of the junction object related list on each of the master object page layouts to have the name of the other master object. For example, you might change the BugCaseAssociations related list to *`Cases`* on the bugs page layout and to *`Bugs`* on the cases page layout. You can further customize these related lists to display fields from the other master object.

To customize the fields that display in the junction object related list on each master object page layout:

1. Edit the page layout of each master object that is related to the junction object. For example, modify the BugCaseAssociations related list for case records by editing the page layout for cases.
2. Edit the properties of the related list you want to modify. For example, on cases the BugCaseAssociations related list was renamed to Bugs, so select the Bugs related list.
3. Add the fields to display in the related list. You can add fields from the junction object itself, but more importantly, you can add fields from the other master object.

   Each field is prefixed with its object name in the popup window. In the related list itself, only fields from the junction object are prefixed with the object name; fields from the other master object are not.

Note: The junction object related list does not include an icon on the master record's detail pages because the junction object does not have a custom tab. If you make a tab for the junction object, the icon is included.

Customizing Reports for Many-to-Many Relationships

Many-to-many relationships provide two standard report types that join the master objects and the junction object. The report types are:

- "Primary master with junction object and secondary master" in the primary master object's report category.
- "Secondary master with junction object and primary master" in the secondary master object's report category.

The order of the master objects in the report type is important. The master object listed first determines the scope of records that can be displayed in the report.

You can create custom reports based on these standard report types. In addition, you can create custom report types to customize which related objects are joined in the report.

SEE ALSO:

- Object Relationships Overview
- Considerations for Relationships
- Define a Custom Object
- Find Object Management Settings
- Find Object Management Settings

Considerations for Relationships

Review the following considerations before creating relationships between objects:

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Relationship Limits

Each custom object can have up to two master-detail relationships and many lookup relationships. Each relationship is included in the maximum number of custom fields allowed.

Converting Relationships

You can convert a master-detail relationship to a lookup relationship as long as no roll-up summary fields exist on the master object.

You can convert a lookup relationship to a master-detail relationship, but only if the lookup field in all records contains a value.

Self Relationships

You can create a relationship from an object to itself, but it must be a lookup relationship, and a single record can't be linked to itself. However, a record can indirectly relate to itself. For example, the Holiday Promotion campaign can have the Direct Mail campaign selected in the lookup relationship, and the Direct Mail campaign can have the Holiday Promotion campaign selected in the lookup relationship.

You can't create a many-to-many self relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Icons for Custom Related Lists

The icon you select for the associated custom tab also displays in any custom related list you create based on a relationship.

Custom related lists do not include an icon if they are based on a relationship with a custom object that does not have a custom tab.

Master-Detail Relationships

To create multilevel master-detail relationships, you need the “Customize Application” user permission.

When you define a master-detail relationship, the custom object on which you are working is the “detail” side. Its data can appear as a custom related list on page layouts for the other object.

By default, records can’t be reparented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the `Allow reparenting` option in the master-detail relationship definition.

You can have up to three custom detail levels.

Standard objects can't be on the detail side of a custom object in a master-detail relationship.

An object can appear once in multilevel master-detail relationships. For example, a subdetail object in one multilevel master-detail relationship can't also be the owner of the master object in another multilevel master-detail relationship. A subdetail object can't also be the master object of the subdetail object's detail object.

Multilevel master-detail relationships do not support division transfers.

You can't create a master-detail relationship if the custom object already contains data. You can, however, create the relationship as a lookup and then convert it to master-detail if the lookup field in all records contains a value.

Converting relationships from lookup to master-detail, or from master-detail to lookup behaves the same as for two-object master-detail relationships. That is, the two linked objects in the detail-subdetail1, or subdetail1-subdetail2 relationship have the same conversion limits as the master-detail relationship.

Roll-up summary fields work as in two-object master-detail relationships. A master can roll up fields on detail records; however, it can't directly roll up fields on subdetail records. To achieve this, the detail record must have a roll-up summary field for the field on the subdetail record, allowing the master to roll up from the detail's roll-up summary field.

You can use multilevel master-detail relationships in custom report types. The `Allow Reports` checkbox must be checked when you create the custom object. Custom report types created for multilevel master-detail relationships count towards the organizations custom report type limit and no reports are generated if this limit is exceeded.

Custom junction objects can't have detail objects. That is, a custom junction object can't become the master object in a multilevel master-detail relationship.

You can't delete a custom object if it is on the master side of a master-detail relationship. If you delete a custom object that is on the detail side of a master-detail relationship, the relationship is converted to a lookup relationship.

Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it, and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later, separately, delete its master record, you cannot undelete the detail record, as it no longer has a master record to relate to.

As a best practice, don't exceed 10,000 child records for a master-detail relationship.

Many-to-Many Relationships

Junction object records are deleted when either associated master record is deleted and placed in the Recycle Bin. If both associated master records are deleted, the junction object record is deleted permanently and can't be restored.

Sharing access to a junction object record is determined by a user's sharing access to both associated master records and the `Sharing Setting` option on the relationship field. See Custom Object Security on page 107. For example, if the sharing setting on both parents is Read/Write, then the user must have Read/Write access to *both* parents in order to have Read/Write access to the junction object. If, on the other hand, the sharing setting on both masters is Read-Only, a user with Read-Only rights on the master records would have Read/Write access to the junction object.

In a many-to-many relationship, a user can't delete a parent record if there are more than 200 junction object records associated with it *and* if the junction object has a roll-up summary field that rolls up to the other parent. To delete this object, manually delete junction object records until the count is fewer than 200.

The first master-detail relationship you create on your junction object becomes the *primary* relationship. This affects the following for the junction object records:

- Look and feel: The junction object's detail and edit pages use the color and any associated icon of the primary master object.
- Record ownership: The junction object records inherit the value of the `Owner` field from their associated primary master record. Because objects on the detail side of a relationship do not have a visible `Owner` field, this is only relevant if you later delete both master-detail relationships on your junction object.
- Division: If your organization uses divisions to segment data, the junction object records inherit their division from their associated primary master record. Similar to the record ownership, this is only relevant if you later delete both master-detail relationships.

The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

Roll-up summary fields that summarize data from the junction object can be created on both master objects.

Formula fields and validation rules on the junction object can reference fields on both master objects.

You can define Apex triggers on both master objects and the junction object.

A junction object can't be on the master side of another master-detail relationship.

You can't create a many-to-many self relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Lookup Relationships

If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:

- `Clear the value of this field` This is the default. Clearing the field is a good choice when the field does not have to contain a value from the associated lookup record.
- `Don't allow deletion of the lookup record that's part of a lookup relationship` This option restricts the lookup record from being deleted if you have any dependencies, such as a workflow rule, built on the relationship.
- `Delete this record also` Available only if a custom object contains the lookup relationship, not if it's contained by a standard object. However, the lookup object can be either standard or custom. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data.

  Warning: Choosing `Delete this record also` can result in a *cascade-delete*. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted *even if they don't have access to the records*. To prevent records from being accidentally deleted, cascade-delete is disabled by default. Contact Salesforce to get the cascade-delete option enabled for your organization.

  Cascade-delete and its related options are not available for lookup relationships to business hours, community, lead, price book, product, or user objects.

In a chain of lookup relationships, these behaviors work independently on each target field at each level. Say, for example, field A is the target lookup of field B, which in turn is the target lookup of field C. You can have a delete restriction on A and none on B, which means that A can't be deleted but B can. Once B is deleted, the relationship between A and B no longer exists and C will hold an empty value for the lookup.

In a multilevel lookup relationship, these options might conflict. For example, in the scenario where field A is the target lookup of field B, which in turn is the target lookup of field C, you might specify that A can delete B, but B cannot be deleted because it's in a relationship with C. If you try to delete A, you'll get an error saying that B can't be deleted because it's linked to C.

If the parent record in a lookup relationship is deleted, the field history tracking for the child record does not record the deletion. For example, if a parent account is deleted, the Account History related list for the child account does not show the deletion.

Relationships on External Objects

Lookup, external lookup, and indirect lookup relationships have some special behaviors and limitations.

- Only lookup, external lookup, and indirect lookup relationships are available for external objects. No other relationship types are supported.
- Depending on the availability of the external system, related lists of child external objects may load slowly when users view the parent record detail pages.
- Relationships that involve external objects allow users to create child records from the record detail pages of parent records. However, the relationship field on each new child record isn't automatically populated to identify the parent record.

- Syncing doesn't create relationship fields on the external objects in your Salesforce org. However, you can change the field type of a sync-created custom field to Lookup Relationship, External Lookup Relationship, or Indirect Lookup Relationship. Changing the field type of an existing custom field is simpler and more efficient than manually creating a relationship field on the external object.

  For example, suppose that the external system has a foreign key relationship. Syncing the related tables creates a text field in your org for the external column that identifies the foreign keys. To reflect the foreign key relationship within your org, change the field type of that text field to External Lookup Relationship.

- A relationship field is a type of custom field. Therefore, like all custom fields on an external object, relationship fields can be overwritten when you sync the external object. See the sync considerations for each Salesforce Connect adapter that you use.
- Cascade-delete isn't available for external object relationships.
- In Salesforce Classic only, external lookup and indirect lookup relationship fields don't display the expected names of parent records.
  - An external lookup field displays either the parent object ID or the value of the parent object's External ID standard field. The latter appears by default. If, however, a custom field on the parent object has the Is Name Field attribute, the parent object ID is displayed.
  - An indirect lookup field displays the value of the target field on the parent object. To find related records, target field values are matched against the values of the indirect lookup relationship field on the child object. The target field, which has the `External ID` and `Unique` attributes, is selected when an indirect lookup field is created.
- When a user tries to edit an external lookup or indirect lookup relationship field, a lookup dialog isn't available for selecting the parent record.
  - To edit an external lookup relationship field, manually enter the value of the `External ID` standard field for the parent record.
  - To edit an indirect lookup relationship field, manually enter the value of the target field of the parent record. The target field is the custom field with `External ID` and `Unique` attributes that was selected when the indirect lookup relationship was created. To determine related records, Salesforce matches target field values against the values of the indirect lookup relationship field on the child object.
- With external lookup and indirect lookup relationships, the parent record appears as a clickable link in the relationship field on the child record. If the child record is viewed by a user who doesn't have access to the parent record, the parent record appears in the relationship field as plain text instead of a link.
- Lookup filters aren't available for external lookup relationship fields.
- Indirect lookup relationship fields can be created on external objects only.
- Only objects that have a custom field with the `External ID` and `Unique` attributes are available as parent objects in indirect lookup relationships. If you don't see the desired object when you create an indirect lookup relationship field, add a custom unique, external ID field to that object.
- If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID**, **Unique**, and **Treat "ABC" and "abc" as different values (case sensitive)**.

Impact of Relationships on Reports

The type of relationship you create affects which standard report types are available and how they are categorized. These report types determine which related objects can be included in the report:

- Lookup relationships allow data from the two related objects to be joined in one report.
- Master-detail relationships allow data from three objects to be joined in one report: the master object, the detail object, plus one other lookup object. If the detail object has multiple lookup relationships, a separate report type is available based on each lookup.

- Many-to-many relationships provide two standard report types that join the master objects and the junction object. The report types are:
  - "Primary master with junction object and secondary master" in the primary master object's report category.
  - "Secondary master with junction object and primary master" in the secondary master object's report category.

  The order of the master objects in the report type is important. The master object listed first determines the scope of records that can be displayed in the report.

The reporting impact of each relationship type is summarized in the following table:

Relationship Type	Standard Report Types	Report Type Category
Lookup	Object by itself Object with first lookup Object with second lookup Object with third lookup	Based on the object
Master-Detail	Master object by itself Master object with detail object Master object with detail object and first lookup Master object with detail object and second lookup Master object with detail object and third lookup	Master object
Many-to-Many	Primary master object by itself Secondary master object by itself Primary master object with junction object and secondary master object Secondary master object with junction object and primary master object	Primary master object and Secondary master object

Custom report types give you more flexibility to join data from multiple objects, including lookups as well as master-detail relationships.

Important: Converting a relationship from lookup to master-detail or vice versa can cause existing custom reports to become unusable due to the different standard report types available for each type of relationship. We recommend that you test your custom reports immediately after converting the relationship type. If you revert your relationship back to the original type, the reports are restored and become usable again.

SEE ALSO:

Object Relationships Overview

Create a Many-to-Many Relationship

External Object Relationships

Create Record Types for Custom Objects

Create record types for a custom object to display different picklist values and page layouts to different users based on their profiles.

1. From the object management settings for a custom object, go to Record Types.
2. Click **New** in the Record Types related list.
3. Choose `Master` from the `Existing Record Type` drop-down list to copy all available picklist values, or choose an existing record type to clone its picklist values.

   Note: When you create a new record type without cloning an existing one, the new record type automatically includes the master picklist values for both standard and custom picklists. You can then customize the picklist values for the record type.

4. Enter a `Record Type Label` that's unique within the object.
5. Enter a description.
6. Select `Active` to activate the record type.
7. Select `Enable for Profile` next to a profile to make the record type available to users with that profile. Select the checkbox in the header row to enable it for all profiles.

   Tip: If each profile is associated with a single record type, users will never be prompted to select a record type when creating new records.

8. For enabled profiles, select `Make Default` to make it the default record type for users of that profile. Select the checkbox in the header row to make it the default for all profiles.
9. Click **Next**.
10. Choose a page layout option to determine what page layout displays for records with this record type:
    - To apply a single page layout for all profiles, select `Apply one layout to all profiles` and choose the page layout from the drop-down list.
    - To apply different page layouts based on user profiles, select `Apply a different layout for each profile` and choose a page layout for each profile.
11. Click **Save** to edit the values of the standard and custom picklists available for the record type, or click **Save and New** to create another record type.

EDITIONS

Available in: Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create record types for custom objects:
- "Customize Application"

SEE ALSO:

Find Object Management Settings

Customize Search Layouts for Custom Objects

You can customize which custom object fields display for users in search results, lookup dialogs, and the key lists on custom tab home pages. You can specify a different set of fields to show in each search layout. The settings apply to all users in your organization.

You can also customize which buttons display in custom list views and search results. You can hide a standard list view button or display a custom button. Standard buttons aren't available on search result layouts. To display a custom button, create the custom button, giving it the `List Button Display Type`.

1. From the object management settings for a custom object, go to Search Layouts.
2. Click **Edit** next to the search layout that you want to modify.

   Note: The fields you select for the key lists on the custom tab home page are also used as the columns displayed in the default All list view and as the default columns for any new list views. To view different columns, customize the list view itself.

3. Move fields between Available Fields and Selected Fields.
   - To customize which fields display in the layout, select one or more fields and click **Add** or **Remove**.
   - To sort fields in the layout, select one or more fields in Selected Fields and click **Up** or **Down**.
   - To select multiple fields individually, use CTRL+click.
   - To select multiple fields as a group, use SHIFT+click.
4. Click **Save**.

SEE ALSO:

Customize Search Layouts

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To define search layouts for custom objects:
- "Customize Application"

Custom Object Security

Learn how security settings work together so you can control access to your custom objects with great flexibility.

Set custom object security at the following levels

- **Tab**—display the custom tab for the appropriate users based on their user profiles.
- **Object**—set the access users have to create, read, edit, and delete records for each object.
- **Records**—set the default sharing model for all your users. This determines the access users have to custom object records that they do not own.
- **Relationship**—for objects on the detail side of a master-detail relationship, specify the sharing access that users must have to the master record in order to create, edit, or delete the associated detail records. This is specified in the `Sharing Setting` attribute of the master-detail relationship field on the detail object.
- **Fields**—set the level of access users have to fields on your custom object page layout.

These requirements apply to custom objects with no master-detail relationship.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Tabs are not available in **Database.com**.

Action	Required Privileges
Create a record	"Create" permission. The user must have the tab displayed to create a new record from the Create New drop-down list in the sidebar.
View a record	"Read" permission and Public Read Only or Public Read/Write sharing model if not the record owner.
Edit a record	"Edit" permission and Public Read/Write sharing model if not the record owner.
Delete a record	"Delete" permission and must be the record owner or above the record owner in the role hierarchy.

These requirements apply to custom objects that have a master-detail relationship with a standard or custom object.

Action	Required Privileges
Create a record	"Create" permission and either read or read/write access to the related master record, depending on the value of the `Sharing Setting` attribute of the master-detail relationship field on the detail object.
View a record	"Read" permission and read access to the related master record. If the record has two master records in a many-to-many relationship, the user must have read access to both master records.
Edit a record	"Edit" permission and either read or read/write access to the related master record, depending on the value of the `Sharing Setting` attribute of the master-detail relationship field on the detail object.
Delete a record	"Delete" permission and either read or read/write access to the related master record, depending on the value of the `Sharing Setting` attribute of the master-detail relationship field on the detail object. When a user deletes a record that has related custom object records, all related custom object records are deleted regardless of whether the user has delete permission to the custom object.

Delegated administrators can manage nearly every aspect of specified custom objects, but they cannot create or modify relationships on the object or set organization-wide sharing defaults.

Notes on Enabling Activities for Custom Objects

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- If you enable activities when creating a custom object, the activity related lists are added to the default page layout automatically. If you enable activities later, after the custom object already exists, you must add the related lists to the page layout manually.
- Disabling activities for a custom object does not delete existing activity records. However, activity related lists are removed from custom object pages, and reports containing activities and the custom object are deleted.
- If a custom object has a master-detail relationship with accounts, the custom object's activities roll up to the account and cause the account's `Last Activity` date to be updated. For custom objects related to other types of records, the activities do not roll up.
- The ability to send emails or create mail merge documents is available for activities on custom objects. The email must be sent to a contact or lead.
- When you change the ownership of a custom object record, any open activities related to that custom object are also transferred to the new record owner.
- You cannot disable activity tracking for a custom object if any workflow tasks are associated with that custom object, or if tasks and events are child data sets of the custom object in a mobile configuration.
- Custom object records can only be associated with a call log in Salesforce CRM Call Center if activities are enabled for the object.

SEE ALSO:

Define a Custom Object

Managing Your Custom Objects and Fields With Schema Builder

Schema Builder

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

Schema Builder provides a dynamic environment for viewing and modifying all the objects and relationships in your app. This greatly simplifies the task of designing, implementing, and modifying your data model, or schema.

You can view your existing schema and interactively add new custom objects, custom fields, and relationships, simply by dragging and dropping. Schema Builder automatically implements the changes and saves the layout of your schema any time you move an object. This eliminates the need to click from page to page to find the details of a relationship or to add a new custom field to an object in your schema.

Schema Builder provides details like the field values, required fields, and how objects are related by displaying lookup and master-detail relationships. You can view the fields and relationships for both standard and custom objects.

Schema Builder is enabled by default and lets you add the following to your schema:

- Custom objects
- Lookup relationships
- Master-detail relationships
- All custom fields except: Geolocation

Note: You can't export your schema from Schema Builder (for example, to use the schema in another org).

SEE ALSO:

How Do I Access Schema Builder?

Custom Field Types

How Do I Access Schema Builder?

From Setup, enter *Schema Builder* in the Quick Find box, then select **Schema Builder**.

When working with Schema Builder:

- Click an object and move it to any space on the canvas. Schema Builder saves the layout of your schema any time you move an object.
- Click **Auto-Layout** to sort the layout of the objects in your schema.

  Important: When you click **Auto-Layout**, you can't undo it.

- Click **View Options** to:
  - **Display Element Names** if you prefer system names, or **Display Element Labels** if you prefer text values.
  - **Show/Hide Relationships**
  - **Show/Hide Legend**
- The **Elements** tab lets you drag and drop new custom objects and fields onto the canvas.
  - To create a custom object, see Creating Objects with Schema Builder.
  - To create a custom field, see Creating Fields with Schema Builder.
- The Objects tab lets you select objects to display on the canvas.
  - Click the drop-down list in the sidebar to filter your list of objects:
    - All Objects
    - Selected Objects
    - Standard Objects
    - Custom Objects
    - System Objects

      Note: Objects created outside of Schema Builder, such as through an app or the API, don't automatically display on the canvas. Select the checkbox for the object created outside Schema Builder to display it on the canvas.
  - To search for an object, type its name in the **Quick Find** box.
  - Hover over an object in your list of objects and click 🔍 to find it on the canvas.
- Hover over relationship lines to show relationship details such as lookup and master-detail relationships. Click the name of the object to find it on the canvas. You can hide relationships if your schema is taking too long to load.
- To view the details of a field in a new window, right-click the element name or label and select **View Field in New Window**.
- To edit properties of a custom field, right-click the element name or label and select **Edit Field Properties**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

USER PERMISSIONS

To view objects in Schema Builder:

- "Customize Application"

- To manage permissions of a custom field, click the element name or label and select **Manage Field Permissions**. Use the dialog box that appears to manage the field's visibility and writeability for all standard and custom profiles. By default, the field-level security for custom fields is set to visible and editable for internal profiles—those not cloned from Partner User or Customer Portal Manager. Fields that are not normally editable, such as formulas and roll-up summary fields, are visible and read only.
- Click  to:
  - **Hide Object on Canvas**
  - **View Object** detail in a new window
  - **View Page Layouts** detail in a new window
- For objects with many fields (Lead or Campaign, for example), click **Show More Fields** to display all the fields.
- To zoom in, click  . To zoom out, click  .

  Note: You can't save the level of zoom when closing Schema Builder.

- To collapse the sidebar, click . To expand it, click .
- The map in the lower right corner shows the overall layout of your objects on the canvas. Click the map to navigate the layout of your objects. To pan across the schema layout while zoomed in, click and hold the canvas while moving the mouse.
- To close the Schema Builder and save the layout of your objects, click **Close**.

Important: If your schema contains many objects and fields, loading can take a long time. Click **Hide Relationships** to improve Schema Builder performance.

SEE ALSO:

Create Objects with Schema Builder
Delete Custom Objects with Schema Builder
Create Fields with Schema Builder
Delete Custom Fields with Schema Builder
Find Object Management Settings

Create Objects with Schema Builder

To create a custom object with Schema Builder:

1. Click the **Elements** tab.
2. Click **Object** and drag it onto the canvas.
3. Enter information to define your object. For a list of object definitions, see Schema Builder Custom Object Definition on page 113.
4. Click **Save**.

SEE ALSO:

Schema Builder Custom Object Definition

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To create new custom objects in Schema Builder:

- "Customize Application"

Create Fields with Schema Builder

To create a custom field with Schema Builder:

1. Click the **Elements** tab.
2. Click a field and drag it onto an object on the canvas.
3. Enter a `Field Label`.

   Salesforce populates `Field Name` using the field label. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

   Ensure that the custom field name and label are unique for that object.

   - If a standard and custom field have identical names or labels, the merge field displays the custom field value.
   - If two custom fields have identical names or labels, the merge field may display an unexpected value.

   If you create a field label called *`Email`* and a standard field labeled `Email` already exists, the merge field may be unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *`Email2`*.
4. Enter a `Description` of the custom field.
5. Enter `Help Text` to detail the purpose and function of a custom field.
6. Enter a `Default Value` to automatically insert a value of a custom field when a new record is created.
7. Depending on the custom field type you choose, enter any remaining field attributes.
8. Click **Save**.

Any field you add through Schema Builder isn't automatically added to the page layout. You will need to edit the page layout to specify where the field should be displayed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To create new fields in Schema Builder:
- "Customize Application"

Delete Custom Objects with Schema Builder

You can delete the custom objects that you no longer need by using Schema Builder.

Schema Builder displays a list of side effects when you try to delete a custom object. Be sure you're ready to accept these side effects before finalizing the deletion. See Delete Custom Objects on page 90 and Manage Deleted Custom Objects on page 92.

1. Click  on the custom object's icon.
2. Select **Delete Object...**. A dialog box displays that explains the side effects of deleting an object. Read this information carefully.
3. If you accept the conditions, check **Yes, I want to delete the custom object.**
4. Click **Delete**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To delete custom objects in Schema Builder:
- "Customize Application"

Delete Custom Fields with Schema Builder

Conveniently avoid "custom field clutter" by using Schema Builder to delete custom fields that you no longer need.

Schema Builder displays a list of side effects when you try to delete a custom field. Be sure you're ready to accept these side effects before finalizing the deletion.

1. Right-click on the custom field.
2. Select **Delete Field...**. A dialog box displays that explains the side effects of deleting a custom field. Read this information carefully.
3. If you accept the conditions, check **Yes, I want to delete the custom field.**
4. Click **Delete**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To delete custom fields in Schema Builder:

- "Customize Application"

Schema Builder Custom Object Definition

Field	Description
`Label`	A name used to refer to the object in any user interface pages.
`Plural Label`	The plural name of the object. If you create a tab for this object, this name is used for the tab.
`Starts with a vowel sound`	If it's appropriate for your organization's default language, check if your label should be preceded by "an" instead of "a".
`Object Name`	A unique name used to refer to the object when using the API. In managed packages, this unique name prevents naming conflicts on package installations. The Object Name field can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	An optional description of the object. A meaningful description helps you remember the differences between your custom objects when you're viewing them in a list.
`Record Name`	The name used in page layouts, list views, related lists, and search results.
`Data Type`	The type of field (text or auto-number) for the record name. Records that have unique IDs instead of names use auto-numbers. An auto-number is a unique number assigned automatically. It is always a read-only field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Field	Description
`Allow Reports`	Makes the data in the custom object records available for reporting purposes. To create reports on custom objects, choose the **Other Reports** report type category, unless the custom object has a relationship with a standard object. When the custom object has a master-detail relationship with a standard object or is a lookup object on a standard object, select the standard object for the report type category instead.
`Allow Activities`	Allows users to associate tasks and scheduled calendar events related to the custom object records.
`Track Field History`	Enables your organization to track changes to fields on the custom object records, such as who changed the value of a field, when it was changed, and what the value of the field was before and after the edit. History data is available for reporting, so users can easily create audit trail reports when this feature is enabled.
`Enable Divisions`	If your organization has divisions enabled, select this option to enable the custom object for divisions. Divisions group records for simplified search results, list views, reports, and other areas within Salesforce. Salesforce adds a `Division` field to the custom object. If the custom object is the master in a master-detail relationship, custom objects on the detail side also get the `Division` field and inherit their division from the master record.
`Available for Customer Portal`	This option makes the custom object available through the Salesforce Customer Portal.
`Namespace Prefix`	In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc are not recognized as unique. Your namespace prefix must be globally unique across all Salesforce organizations. It keeps your managed package under your control exclusively.
`Deployment Status`	Indicates whether the custom object is visible to other users.
`Add Notes & Attachments...`	Allows users to attach notes and attachments to custom object records. This allows you to attach external documents to any object record, in much the same way that you can add a PDF or photo as an attachment to an email. This option is only available when you are creating a new object.

Creating Unique Sets of Data

Custom Settings

Use custom settings to create custom sets of data, or to create and associate custom data for an org, profile, or user.

Custom settings are similar to custom objects in that they let you customize org data. Unlike custom objects which have records based on them, custom settings let you utilize custom data sets across your org, or distinguish particular users or profiles based on custom criteria.

Custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data can then be used by formula fields, validation rules, flows, Apex, and the SOAP API

Note: If you're thinking of using List Custom Settings, consider using Custom Metadata Types instead. Unlike with List Custom Settings, you can migrate the records of Custom Metadata Types using Packages or Metadata API tools.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

Packages are not available in **Database.com**.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:
- "Customize Application"

There are two types of custom settings:

List Custom Settings
A type of custom setting that provides a reusable set of static data that can be accessed across your organization. If you use a particular set of data frequently within your application, putting that data in a list custom setting streamlines access to it. Data in list settings does not vary with profile or user, but is available organization-wide. Examples of list data include two-letter state abbreviations, international dialing prefixes, and catalog numbers for products. Because the data is cached, access is low-cost and efficient: you don't have to use SOQL queries that count against your governor limits.

Hierarchy Custom Settings
A type of custom setting that uses a built-in hierarchical logic that lets you "personalize" settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or "lowest," value. In the hierarchy, settings for an organization are overridden by profile settings, which, in turn, are overridden by user settings.

The following examples illustrate how you can use custom settings:

- A shipping application requires users to fill in the country codes for international deliveries. By creating a list setting of all country codes, users have quick access to this data without needing to query the database.
- An application calculates and tracks compensation for its sales reps, but commission percentages are based on seniority. By creating a hierarchy setting, the administrator can associate a different commission percentage for each profile in the sales organization. Within the application, one formula field can then be used to correctly calculate compensation for all users; the personalized setting at the profile level inserts the correct commission percentage.
- An application displays a map of account locations, the best route to take, and traffic conditions. This information is useful for sales reps, but account executives only want to see account locations. By creating a hierarchy setting with custom checkbox fields for route and traffic, you can enable this data for just the "Sales Rep" profile.

To create and use custom settings:

1. Create the custom setting.
2. Add fields to the custom setting.
3. Add data and set the access level for the custom setting data.
4. Reference the custom setting data in your application, using formula fields, validation rules, Apex, or the SOAP API.

You can also include a custom setting in a package. The visibility of the custom setting in the package depends on the `Visibility` setting.

Note: Only custom settings definitions are included in packages, not data. If you need to include data, you must populate the custom settings using a standard Apex or API script run by the subscribing organization after they have installed the package.

SEE ALSO:

View Custom Settings

Custom Settings Limits and Considerations

Accessing Custom Settings

You can access custom settings from formula fields, validation rules, Apex, and the SOAP API. Some sample code segments are provided below.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

Packages are not available in **Database.com**.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:
- "Customize Application"

Formula Fields

Formula fields only work for hierarchy custom settings; they can't be used for list custom settings.

```
{!$Setup.CustomSettingName__c.CustomFieldName__c}
```

Apex

Apex can access both custom setting types.

Note: If **Privacy** for a custom setting is `Protected` and the custom setting is contained in a managed package, the subscribing organization cannot edit the values or access them using Apex.

Samples for List Custom Settings

When you add data to a custom setting, you must name each set of data. Then you can distinguish between the sets of data by the data set name. The following returns a map of custom settings data. The `getAll` method returns values for all custom fields associated with the list setting.

```
Map<String_dataset_name, CustomSettingName__c> mcs = CustomSettingName__c.getAll();
```

The following example uses the `getValues` method to return all the field values associated with the specified data set. This method can be used with both list and hierarchy custom settings, using different parameters.

```
CustomSettingName__c mc = CustomSettingName__c.getValues(data_set_name);
```

Samples for Hierarchy Custom Settings

The following example uses the `getOrgDefaults` method to return the data set values for the organization level:

```
CustomSettingName__c mc = CustomSettingName__c.getOrgDefaults();
```

The following example uses the `getInstance` method to return the data set values for the specified profile. The `getInstance` method can also be used with a user ID.

```
CustomSettingName__c mc = CustomSettingName__c.getInstance(Profile_ID);
```

SOAP API

Custom settings that have **Privacy** defined as `Public` are exposed to the API in the same way custom objects are exposed.

Note: If **Privacy** is defined as `Protected`, and the custom setting is contained in a managed package, the custom setting is not accessible using the API in either the developer organization or a subscribing organization.

Use any tool with API access to perform query or profile-permission-setting operations.

Note: You can also access custom settings data through a Standard Object Query Language (SOQL) query, but this method doesn't make use of the application cache. It's similar to querying a custom object.

SEE ALSO:

Custom Settings

Define Custom Settings

To create or edit a custom setting:

1. From Setup, enter *Custom Settings* in the `Quick Find` box, then select **Custom Settings**.
2. Click **New** to create a new custom setting, click **Edit** next to the name of a custom setting, or click **Edit** while viewing the details of a custom setting.

   Note: A icon indicates that the custom setting is in an installed managed package. You can't edit or delete a custom setting installed from a managed package.

3. Define the following:
   - `Label`—Enter the label displayed in the application.
   - `Object Name`—Enter the name to be used when the custom setting is referenced by formula fields, validation rules, Apex, or the SOAP API.

     Note: Salesforce recommends using ASCII for the `Object Name`. The name can't exceed 38 ASCII characters. If you use double byte, there are additional limits on the number of characters allowed.

   - `Setting Type`—Select a type of List or Hierarchy. The List type defines application-level data, such as country codes or state abbreviations. The Hierarchy type defines personalization settings, such as default field values, that can be overridden at lower levels in the hierarchy.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

Packages are not available in **Database.com**.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:
- "Customize Application"

Important: After you save a custom setting, you cannot change this value.

- `Visibility`—Select a visibility of Protected or Public.
  - Protected—If the custom setting is contained in a managed package, subscribing organizations can't see the custom setting: it doesn't display as part of the package list. In addition, subscribing organizations can't access the custom setting using either Apex or the API, however, developer organizations can. If the custom setting is contained in an unmanaged package, the custom setting is available through the Enterprise WSDL like any custom object (as if the `Visibility` was Public.)
  - Public—The custom setting is available through the Enterprise WSDL like any custom object. You can package custom settings defined as public. The subscribing organizations can edit the values, as well as access them using Apex and the API, regardless of the type of package (either managed or unmanaged).

  Important: After you save a custom setting, you cannot change this value.

4. Enter an optional description of the custom setting. A meaningful description will help you remember the differences between your custom settings when you're viewing them in a list.
5. Click **Save**.

   Note: Only custom settings definitions are included in packages, not data. If you need to include data, you must populate the custom settings using a standard Apex or API script run by the subscribing organization after they have installed the package.

After you create a custom setting, you must also add fields to the custom setting.

SEE ALSO:

Add Custom Settings Fields

Add Custom Settings Data

View Custom Settings

After you create a custom setting, you can view the details of the custom setting, manage the custom setting, and add fields.

From Setup, enter *`Custom Settings`* in the `Quick Find` box, then select **Custom Settings**, then click the name of the custom setting you'd like to view. While viewing a custom setting, you can:

- Click **Edit** to make changes to a custom setting.
- Click **Delete** to delete a custom setting.

  Note: A icon indicates that the custom setting is in an installed managed package. You can't edit or delete a custom setting installed from a managed package.

- Click **Manage** to add data to a custom setting.

In addition, click **New** to add fields to the custom setting.

SEE ALSO:

Define Custom Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

Packages are not available in **Database.com**.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:

- "Customize Application"

Add Custom Settings Fields

After you define custom settings, add fields to them. The custom fields contain the data used by the custom setting.

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.
2. Click the custom setting that you want to add fields to. (If you just created the custom setting, the Custom Setting Detail page appears.)
3. Click **New**.
4. Select a field type and click **Next**.

   Note: Record size is based on the maximum field size of each field type, not the actual storage that's used in each field. When adding fields to a custom setting definition, use the appropriate type and specify a length that doesn't exceed what's needed for your data. This action helps you avoid reaching the cached data limit. For example, if you create a US social security number (SSN) field, select the Text data type and specify a length of 9. If instead you selected a Text Area data type, the field would add 255 characters to the usage count for each record, regardless of the number of characters entered.

5. Enter the details for the field.
6. Confirm the information, and then click **Save** or **Save & New**.

After you add fields, you need to add data, and for hierarchy custom settings, specify the access level.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

Packages are not available in **Database.com**.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:
- "Customize Application"

SEE ALSO:

Add Custom Settings Data

Manage Custom Settings Data

After defining custom settings and adding fields, populate the fields:

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.
2. Click **Manage** next to a custom setting, or from the detail page for a custom setting.
3. Provide or change values for the custom setting.
   - If you are managing a list setting:
     - Click **New** to add data to the fields.
     - Click **Edit** next to the name of an existing set of data to change the name of the data set or to change the data.
     - Click **Del** next to the name of an existing set of data to delete the data set.
   - If you are managing a hierarchy setting, decide where in the permission hierarchy you want to add default data (organization, profile, or user).

     To add default data at the organization level, click **New** in the Default Organization Level Value section. If data has already been defined for the organization, you can only edit or delete it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:
- "Customize Application"

To add default data at the profile or user level, click **New** in the lower section of the page, near the Setup Owner.

After you have defined data, you can:

- Click **Edit** in the Default Organization Level Value section to change the default data set at the organization level, or **Delete** to delete it (this is only for hierarchical custom settings.)
- Click **View** next to the name of an existing set of data to view the data (this is only for hierarchical custom settings.)
- Click **Edit** next to the name of an existing set of data to change the name of the data set or to change the data.
- Click **Del** next to the name of an existing set of data to delete the data set.

SEE ALSO:

Custom Settings Limits and Considerations

Add Custom Settings Data

Add Custom Settings Data

After you define your custom settings and add fields, you need to populate the fields with data.

You can define one or more data sets. For list custom settings, each data set is named and can be accessed by that name using Apex, formula fields, and so on.

For custom settings that are hierarchies, the data is accessed based on the access level (user, profile, or organization). The lowest level is used first, which means if you defined a data set at the user level, unless otherwise specified in your application, that data is used. For example, you might want to specify different contact numbers for your application: one for the general user, and one that is only displayed for system administrators.

To add data to custom setting fields:

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**, then click **Manage** next to a custom setting. Or from the detail page for a custom setting, click **Manage**.
2. Click **New** or **Edit** next to an existing data set.
3. Add or change data.

   For custom settings that are lists:

   a. Specify or change the name for the data set. This name is used by Apex, formula fields, and so on.

   b. Enter or change data for all fields.

   c. Click **Save**.

   For custom settings that are hierarchies:

   a. For the default organization level values, enter or change the data for the fields. The default organization location is automatically populated.

   b. For profile or user level values, select either Profile or User from the Location picklist. Enter the name of the profile or user, or use the lookup dialog search. Then enter or change the data for the fields.

   c. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:

- "Customize Application"

Note: For a hierarchy custom setting, you can add only one record for a profile or user. Adding two records for the same profile or user results in an error.

SEE ALSO:

Manage Custom Settings Data

Add Custom Settings Fields

View Custom Settings Data

After you add fields and add data to those fields, you can view the data.

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**, then click **Manage** next to a custom setting that has already been defined. Or from the detail page for a custom setting, click **Manage**.
2. Click **View** next to the data set you want to view (this is only for hierarchical custom settings).

SEE ALSO:

Add Custom Settings Fields

Add Custom Settings Data

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:

- "Customize Application"

Custom Settings Limits and Considerations

When working with custom settings, be aware of the following considerations and limits on the amount of cached data.

- The total amount of cached data allowed for your org is the **lesser** of these two values:
  - 10 MB
  - 1 MB multiplied by the number of full-featured user licenses in your org

  For example, if your org has three full licenses, you have 3 MB of custom setting storage. If your org has 20 full licenses, you have 10 MB of storage.

  Each certified managed package gets a separate limit in addition to your org limit. For example, let's say your org has two certified managed packages installed and your organization has three full licenses. Each certified managed package can have 3 MB of custom setting storage in addition to your org's 3-MB custom setting storage limit.
- You can add up to 300 fields per custom setting, unless your field limit for custom objects is lower than 300. If your custom objects field limit is lower than 300, your field limit for custom settings is equal to your custom objects field limit.
- You can't share a custom setting object or record.
- No owner is assigned when a custom setting is created, so the owner can't be changed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions.

USER PERMISSIONS

To manage, create, edit, and delete custom settings:

- "Customize Application"

- Custom settings are a type of custom object. Each custom setting counts against the total number of custom objects available for your organization.
- Accessing an undeleted custom setting in a formula field results in an error if the user doesn't have the "Customize Application" permission. To prevent this error, redeploy this custom setting to the organization. Alternatively, delete this custom setting, re-create it with the same name and data, and then delete and re-create all formula fields that use this setting.
- If a cross-object formula references a currency field from a custom setting, this field value isn't converted to the currency of the record containing the formula. The result of the formula could be inaccurate if the custom setting field's currency and the record's currency are different.

To see how much custom settings data your organization is using, from Setup, enter `Custom Settings` in the `Quick Find` box, then select **Custom Settings**. For each custom setting, this page lists the size of one record, the number of records created, and the total size used for each custom setting.

Record size is based on the maximum field size of each field type, not the actual storage that's used in each field. When adding fields to a custom setting definition, use the appropriate type and specify a length that doesn't exceed what's needed for your data. This action helps you avoid reaching the cached data limit. For example, if you create a US social security number (SSN) field, select the `Text` data type and specify a length of 9. If instead you selected a `Text Area` data type, the field would add 255 characters to the usage count for each record, regardless of the number of characters entered.

SEE ALSO:

Custom Settings

Working with Custom Fields

Customize Fields

Customize standard and custom fields to tailor your org to your own unique requirements.

You can:

- Modify some aspects of standard fields
- Change or add values to standard and custom picklist fields
- Define dependency rules between fields
- Create custom fields to capture additional information
- Create formula fields that automatically calculate values based on the contents of other fields
- Define default values for custom fields
- Define validation rules for your fields
- Make a field required
- Set fields to track changes, including the date, time, nature of the change, and who made the change
- Create page layouts to control the display of fields
- Set field-level security to control access to fields
- Create or modify field sets

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Standard Fields and Page Layouts are not available in **Database.com**

USER PERMISSIONS

To create or change custom fields:

- "Customize Application"

Customize Standard Fields

You can customize several aspects of standard fields, such as the values in picklists, the format for auto-number fields, tracking field history, lookup filters on relationship fields, and field-level help.

Tip: You can't delete standard fields, but you can remove them from your page layouts.

1. Navigate to the fields page for your object.
2. Click the field label.
3. To add custom help text, click **Edit**.
4. On the field's information page, you can—depending on your Edition—set field-level security, view accessibility settings, and configure validation rules.

   You can also do more, depending on the field's type. For example, if the field is a picklist, you can add, delete, and reorder its values, and set dependencies. You can't increase the field length of a standard field. If you need a longer text area, consider creating a custom field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To change standard fields:
- "Customize Application"

Beyond the Basics

What does that `Indexed` checkbox mean on a field, and how did it get there?

If a field is indexed, you can use sidebar search or advanced search to find values in the field. Having a field indexed can also speed up other operations on the field, such as reporting. Check out this blog post to find out more: Know Thy Salesforce Field Indexes for Fast Reports, List Views, and SOQL.

SEE ALSO:

Custom Fields
Add or Edit Picklist Values
Rename Object, Tab, and Field Labels
Getting Started with Field-Level Help
Lookup Filters

Modify Standard Auto-Number Fields

The unique identifiers for solutions, cases, and contracts are standard auto-number fields. Each record is assigned a unique number with a specified format upon creation. You can modify the format and numbering for these auto-number fields.

1. From the management settings for the object whose field you want to modify, go to the fields area.
2. Click **Edit** next to the name of the field.
3. Enter a `Display Format` to control such formatting details as the minimum number of leading zeros as well as any prefix or suffix for the number. See Custom Field Attributes on page 134.

   Format changes do not affect existing records; they are applied only to new records.
4. Enter the number to be assigned to the next record that is created after you save your changes.
5. Click **Save**.

Warning: Salesforce warns you if the next number you enter is not higher than existing numbers. However, it may be possible to create duplicate numbers if you change the auto-number format multiple times using similar formats each time.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To modify standard auto-number fields:
- "Customize Application"

SEE ALSO:

Custom Field Types

Custom Fields

To tailor your organization, you can add custom fields for each of the tabs and objects that your organization uses.

For the total number of custom fields you can create, see *Salesforce Limits*.

Note: When your org is close to the limit of 800 custom fields and you delete or create fields, field creation can fail. The physical delete process reclaims and cleans fields, making them count temporarily toward the limit. The delete process runs only when the queue is full, so it can take days or weeks to start. In the meantime, the deleted fields are still counted as part of the limit. To request immediate deletion of fields, contact Salesforce Support.

See the following for more information:

- Create Custom Fields
- Edit Custom Fields
- Manage Fields for a Specific Object
- Delete Fields
- Build a Formula Field
- Create a Custom Picklist Field
- Define Dependent Picklists
- Define Default Field Values
- Additional Custom Field Options
- Custom Field Attributes

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Tabs are not available in **Database.com**

USER PERMISSIONS

To create or change custom fields:
- "Customize Application"

- Change the Custom Field Type

Creating and Enhancing Custom Fields

Create Custom Fields

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.

Watch a Demo: How to Create a Custom Field in Salesforce

Want to customize Salesforce so it captures all your business data? This short video walks you through how to create a custom picklist field, from choosing the correct field type to applying field level security.

Before you begin, determine the type of field you want to create.

Note: When your org is close to the limit of 800 custom fields and you delete or create fields, field creation can fail. The physical delete process reclaims and cleans fields, making them count temporarily toward the limit. The delete process runs only when the queue is full, so it can take days or weeks to start. In the meantime, the deleted fields are still counted as part of the limit. To request immediate deletion of fields, contact Salesforce Support.

1. From the management settings for the object you want to add a field to, go to Fields.

   Custom task and event fields are accessible from the object management settings for Activities.

2. Click **New**.

   Tip: On custom objects, you can also set field dependencies and field history tracking in this section.

3. Choose the type of field and click **Next**. Consider the following.
   - Some data types are available for certain configurations only. For example, the `Master-Detail Relationship` option is available for custom objects only when the custom object doesn't already have a master-detail relationship.
   - Custom settings and external objects allow only a subset of the available data types.
   - You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.
   - Relationship fields count towards custom field limits.
   - Additional field types may appear if an AppExchange package using those field types is installed.
   - The `Roll-Up Summary` option is available on certain objects only.
   - Field types correspond to API data types.
   - If your organization uses Shield Platform Encryption, ensure you understand how to encrypt custom fields using the Shield Platform Encryption offering.

4. For relationship fields, associate an object with the field and click **Next**.
5. For indirect lookup relationship fields, select a unique, external ID field on the parent object, and then click **Next**. The parent field values are matched against the values of the child indirect lookup relationship field to determine which records are related to each other.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Custom fields aren't available on Activities in **Group** Edition

Custom settings aren't available in **Professional** Edition

Layouts aren't available in **Database.com**

USER PERMISSIONS

To create or change custom fields:
- "Customize Application"

6. To base a picklist field on a global picklist value set, select the value set to use.
7. Enter a field label.

   Salesforce populates `Field Name` using the field label. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Use the field name for merge fields in custom links, custom s-controls, and when referencing the field from the API.

   Tip: Ensure that the custom field name and label are unique for that object.

   - If a standard and custom field have identical names or labels, the merge field displays the custom field value.
   - If two custom fields have identical names or labels, the merge field may display an unexpected value.

   If you create a field label called *`Email`* and a standard field labeled `Email` already exists, the merge field may be unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *`Email2`*.
8. Enter field attributes and select the appropriate checkboxes to specify whether the field must be populated and what happens if the record is deleted.
9. For master-detail relationships on custom objects, optionally select **Allow reparenting** to allow a child record in the master-detail relationship to be reparented to a different parent record.
10. For relationship fields, optionally create a lookup filter to limit search results for the field. Not available for external objects.
11. Click **Next**.
12. In Enterprise, Unlimited, Performance, and Developer Editions, specify the field's access settings for each profile, and click **Next**.

Access Level	Enabled Settings
Users can read and edit the field.	**Visible**
Users can read but not edit the field.	**Visible** and **Read-Only**
Users can't read or edit the field.	None

Note:

- When you create a custom field, by default the field isn't visible or editable for portal profiles, unless the field is universally required.
- Profiles with "View Encrypted Data" permission are indicated with an asterisk.

13. Choose the page layouts that will display the editable field and click **Next**.

Field	Location on Page Layout
Normal	Last field in the first two-column section.
Long text area	End of the first one-column section.
User	Bottom of the user detail page.
Universally required	Can't remove it from page layouts or make read only.

14. For relationship fields, optionally create an associated records related list and add it to page layouts for that object.

- To edit the related list name on page layouts, click **Related List Label** and enter the new name.
- To add the related list to customized page layouts, select `Append related list to users' existing personal customizations`.

15. Click **Save** to finish or **Save & New** to create more custom fields.

Note: Creating fields may require changing a large number of records at once. To process these changes efficiently, your request may be queued and you may receive an email notification when the process has completed.

SEE ALSO:

External Object Relationships

Find Object Management Settings

Create a Custom Picklist Field

Create custom picklist fields to let your users select values from lists that you define.

Watch a Demo: Custom Fields: Picklists

You can create these types of picklist fields:

- Local picklist—Lets users select a single value from a list that you define. This picklist is unique and had its own set of values.
- Shared picklist—Lets users select a single value from a global picklist value set that you define in Setup. All custom picklist fields that use a global value set inherit its values and can't have additional values.
- Multi-select picklist—Allows users to select more than one picklist value from a list that you define. These fields display each value separated by a semicolon.

Note: Picklist fields aren't available for external objects. You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To create or change custom fields:

- "Customize Application"

1. Go to the fields area of the object you want to create a picklist field for.

2. In the custom fields related list, click **New**.

3. Select **Picklist** or **Picklist (Multi-Select)**, and then click **Next**.

4. Enter a label for the picklist field.

5. To use the values from an existing global picklist, select **Use global picklist value set**. To use values that you create specifically for this picklist, select **Enter values for the picklist, with each value separated by a new line.**

Tip: Provide feedback and suggestions for global picklists in the Global, Restricted Custom Picklists group in the Salesforce Success Community.

6. If you didn't use a global picklist value set, enter picklist values.

Put each value on a separate line. Values can be up to 255 characters long.

7. Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both. You can't change these settings later on an existing picklist field.

If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default.

Note: Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See About Default Field Values on page 167.

8. Choose whether to restrict this picklist's values to an admin-approved list. Selecting **Restrict picklist to the values defined in the value set** prevents users from loading unapproved values through the API.
9. If you're creating a multi-select picklist, enter how many values you want displayed at a time on edit pages. The number of values determines the box height.
10. Enter description or help text if desired, and then click **Next**.
11. Set field-level security for the picklist field, and then click **Next**.
12. Choose the page layouts on which to include the picklist field.
13. Click **Save**.

SEE ALSO:

Create Custom Fields

Add or Edit Picklist Values

Create a Global Picklist Value Set

Create a Global Picklist Value Set

Use global picklist value sets to share values across objects and custom picklist fields, and to restrict the picklists to only the values that you specify.

Note: Give your feedback and suggestions for global picklists in the Global, Restricted Custom Picklists group in the Salesforce Success Community.

A custom picklist is tied to a particular object as a field on the object. Unlike a custom picklist field, a global picklist exists independently as a *global picklist value set*. Its values are shared with any picklist that's based on it.

A global picklist is a restricted picklist by nature. Only a Salesforce admin can add to or modify its values. Users can't add unapproved values, even through the API.

Note: You can have up to 500 global picklists (or picklist value sets) in an org. Each global value set can contain up to 500 values. These limits don't apply to custom picklists that use global picklist value sets.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS

To create or change custom fields:

- "Customize Application"

1. From Setup, enter `Picklists` in the `Quick Find` box, then select **Picklist Value Sets**.
2. Next to Global Value Sets, click **New**.
3. Enter a label for the global value set. This name appears in Setup, and when users create a picklist based on this global value set.
4. To tell users what these values are for, enter a specific description of the global value set. This text appears on the Picklist Value Sets list page in Setup.
5. Enter the values, one per line.
6. Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both. You can't change these settings later.

   If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default.
7. Click **Save**.

Your global value set is ready to be used in custom picklist fields. You can reorder the values by clicking **Reorder**.

To create a picklist that uses a global picklist value set, see Create a Custom Picklist Field.

To see all of the fields where this value set is used, look under Fields Where Used on the global picklist's detail page.

SEE ALSO:

Manage Inactive Picklist Values

Create a Custom Picklist Field

Custom Field Types

When you have data that doesn't match any of the standard fields, your administrator can create a custom field for that data. For example, you might want a `Middle Name` field for contacts.

The first step in creating a custom field is choosing the type of the field. Below is a description of each custom field type. Additional field types may appear if an AppExchange package using those field types is installed.

Type	Description
Auto Number	Automatically assigns a unique number to each record. The maximum length of any auto-number field is 30 characters, 20 of which are reserved for prefix or suffix text. Not available for external objects.
Checkbox	Allows users to check a box, indicating a true or false attribute of a record. When using a checkbox field for a report or list view filter, use "True" for checked values and "False" for unchecked values. The Data Import Wizard and the weekly export tool use "1" for checked values and "0" for unchecked values.
Currency	Allows users to enter a currency amount. The system automatically formats the field as a currency amount. This can be useful if you export data to a spreadsheet application. Not available for external objects. Note: Salesforce uses the round-half-to-even tie-breaking rule for currency fields. For example, 23.5 becomes 24, 22.5 becomes 22, –22.5 becomes –22, and –23.5 becomes –24. Values lose precision after 15 decimal places.
Date	Allows users to enter a date or pick a date from a popup calendar. In reports, you can limit the data by specific dates using any custom date field.
Date/Time	Allows users to enter a date or pick a date from a popup calendar and enter a time of day. They can also add the current date and time by clicking the date and time link next to the field. The time of day includes AM or PM notation. In reports, you can limit the data by specific dates and times using any custom date field.
Email	Allows users to enter an email address of up to 80 characters, which is validated to ensure proper format. If this field is specified for contacts or leads, users can choose the address when clicking **Send**

Type	Description
	an Email. Note that you can't use custom email addresses for mass emails. Note: This field can be encrypted using Shield Platform Encryption.
External Lookup Relationship	An external lookup relationship links a child standard, custom, or external object to a parent external object. When you create an external lookup relationship field, the standard External ID field on the parent external object is matched against the values of the child's external lookup relationship field. External object field values come from an external data source.
Formula	Allows users to automatically calculate values based on other values or fields such as merge fields. Not available for external objects. Note: Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and –12.345 becomes –12.34.
Geolocation	Allows users to specify a location by its latitude and longitude. Geolocation is a compound field that counts toward your org's limits as three custom fields: one for latitude, one for longitude, and one for internal use. Not available for external objects.
Hierarchical Relationship	Creates a hierarchical lookup relationship between users. Allows users to use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.
Indirect Lookup Relationship	An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values come from an external data source.
Lookup Relationship	Creates a relationship between two records so you can associate them with each other. For example, opportunities have a lookup relationship with cases that lets you associate a particular case with an opportunity. • On a standard or custom object, a lookup relationship creates a field that allows users to click a lookup icon and select another record from a popup window. • On an external object, the lookup relationship field references 18-character Salesforce IDs that are stored in an external data

Type	Description
	source. Those IDs are matched against the parent object to determine which records are related to each other. On the parent record, you can display a related list to show all of the records that are linked to it. You can create lookup relationship fields that link to users, standard objects, or custom objects. If a lookup field references a record that has been deleted, Salesforce clears the value of the lookup field by default. Alternatively, you can choose to prevent records from being deleted if they're in a lookup relationship. Lookup relationship fields are not available in Personal Edition. Lookup relationship fields to campaign members are not available; however, lookup relationship fields from campaign members to standard or custom objects are available.
Master-Detail Relationship	Creates a relationship between records where the master record controls certain behaviors of the detail record such as record deletion and security. Not available for standard objects or external objects, although you can create a master-detail relationship field on a custom object that links to a standard object. Master-detail relationships cannot be used with campaign members.
Number	Allows users to enter any number. This is treated as a real number and any leading zeros are removed. Note: Salesforce uses the round half up tie-breaking rule for number fields. For example, 12.345 becomes 12.35 and −12.345 becomes −12.34. Salesforce rounds numbers referenced in merge fields according to the user's locale, not the number of decimal spaces specified in the number field configuration.
Percent	Allows users to enter a percentage number as a decimal—for example, `0.10`. The system automatically converts the decimal to a percentage—for example, `10%`. Note: Values lose precision after 15 decimal places. Also, if you enter a value with more than 15 decimal places and add a percent sign to the number, a runtime error occurs.
Phone	Allows users to enter any phone number. Character limit is 40. Salesforce automatically formats it as a phone number. If you are using Salesforce CRM Call Center, custom phone fields are displayed with the button, allowing click-to-dial

Type	Description
	functionality. Consequently, Salesforce recommends that you do not use a custom phone field for fax numbers. Note: This field can be encrypted using Shield Platform Encryption.
Picklist	Lets users select a single value from a list that you define. Not available for external objects.
Picklist (Multi-select)	Allows users to select more than one picklist value from a list that you define. These fields display each value separated by a semicolon. Not available for external objects.
Roll-Up Summary	Automatically displays the record count of related records or calculates the sum, minimum, or maximum value of related records. The records must be directly related to the selected record and on the detail side of a custom master-detail relationship with the object that contains the roll-up summary field. For example, a custom field called "Total Number of Guests" displays the number of guest custom object records in the Guests related list. Not available for external objects.
Text	Allows users to enter any combination of letters, numbers, or symbols. You can set a maximum length, up to 255 characters. Note: This field can be encrypted using Shield Platform Encryption.
Text (Encrypted)	Allows users to enter any combination of letters, numbers, or symbols that are stored in encrypted form. You can set a maximum length of up to 175 characters. Encrypted fields are encrypted with 128-bit master keys and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact Salesforce. Not available for external objects. Note: This field can be encrypted using Classic Encryption. If your org uses Shield Platform Encryption, use `Text` to create an encrypted text field.
Text Area	Allows users to enter up to 255 characters that display on separate lines similar to a `Description` field.
Text Area (Long)	Allows users to enter up to 131,072 characters that display on separate lines similar to a `Description` field. You can set the length of this field type to a lower limit, if desired. Any length from 256 to 131,072 characters is allowed. The default is 32,768 characters. Note that every time you press `Enter` within a long text area field, a line break and a return character are added to the text. These two characters count toward the 131,072 character

Type	Description
	limit. This data type is not available for activities or products on opportunities. Only the first 254 characters in a rich text area or a long text area are displayed in a report. Note: This field can be encrypted using Shield Platform Encryption.
Text Area (Rich)	With the use of a toolbar, users can format the field content and add images and hyperlinks. The toolbar allows the user to undo, redo, bold, italicize, underline, strike-out, add a hyperlink, upload or link to an image, modify alignment, add a numbered or non-numbered list, indent, and outdent. The maximum field size is 131,072 characters, inclusive of all the formatting and HTML tags. Only the first 254 characters in a rich text area or a long text area are displayed in a report. The maximum size for uploaded images is 1MB. Only `gif`, `jpeg` and `png` file types are supported. Not available for external objects.
URL	Allows users to enter up to 255 characters of any valid website address. When users click on the field, the URL will open in a separate browser window. Note that only the first 50 characters are displayed on the record detail pages. Note: This field can be encrypted using Shield Platform Encryption.

SEE ALSO:

Custom Field Attributes

What's the Difference between Standard Fields and Custom Fields?

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

You can add custom fields to standard and custom objects in Salesforce, allowing you to infinitely customize your organization. Although custom fields are similar to the standard fields that come built-in to Salesforce, there are some differences between standard fields and custom fields.

Custom objects and fields let you tailor which data is stored to fit your organization's needs. Though the Force.com database provides several standard objects and accompanying standard fields, you can easily customize how you track and report on your data.

Custom objects give you the flexibility to store any type of enterprise data that's relevant to your app, by creating objects. For example, if you're building a recruiting app, you can create custom objects called Position and Candidate to track information on job openings and candidates, respectively.

Categorize and track your data even more granularly by using custom fields. Like standard objects, custom objects have fields that define the data for those object records. You can, for example, add a custom number field `Years of Experience` to the custom candidate object or a custom text field `Description` to the custom position object. You can add custom fields to both custom and standard objects.

See the following for more information:

- Create Custom Fields
- Edit Custom Fields
- Manage Fields for a Specific Object
- Build a Formula Field
- Define Default Field Values
- Additional Custom Field Options
- Custom Field Attributes
- Change the Custom Field Type

Custom Field Attributes

A custom field entry consists of several attributes.

Field	Description
# Visible Lines	For long text area fields, set the number of lines to be displayed on edit pages. You can display between 2 and 50 lines (the default is 6 lines). If the text does not fit in the specified number of visible lines, scroll bars appear. Long text area fields are fully displayed on detail pages and printable views.
Calculation Options	Determines how a roll-up summary field is recalculated after its properties change. Choose `Automatic calculation` to recalculate a field the next time it's displayed. Choose `Force a mass recalculation of this field` as a safety net option to force recalculation of the roll-up summary field values.
Child Relationship Name	The name used in API SOQL relationship queries.
Data Type	The data type of a field determines what type of information is in the field. For example, a field with the Number data type contains a positive or negative integer. For more information on data types, see Custom Field Types on page 129.
Decimal Places	For currency, geolocation, number, and percent fields, this field represents the number of digits you can enter to the right of a decimal point. The system rounds the decimal numbers you enter, if necessary. For example, if you enter *`4.986`* in a field with `Decimal Places` set to *`2`*, the number rounds to 4.99.
Default Value	The value to apply when a user creates a record. For checkbox custom fields, choose **Checked** or **Unchecked** as the default value to indicate the default when creating records. Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See About Default Field Values on page 167.
Description	Text that describes the custom field. This description is for administration purposes only and doesn't display to users on record detail and edit pages that include the field.
Display Format	For auto-number fields, enter a `Display Format` to control formatting details such as the minimum number of leading zeros and any prefix or suffix for the number. Begin by entering the required minimum **{0}** as a placeholder for the auto-number without any leading zeros. Add any prefix to your number before this placeholder

Field	Description
	and insert any suffix text after the placeholder. Insert any date prefixes or suffixes in the form of **{YY}**, **{YYYY}**, **{MM}**, or **{DD}**, which always represent the create date of the record. For information on using auto-number formats when entering your `Display Format`, see Auto-Number Formatting Examples on page 139.
Encrypted	If checked, this custom field is encrypted using Shield Platform Encryption. Note: This page is about Shield Platform Encryption, not Classic Encryption. What's the difference?
External Column Name	Available on external objects only. Maps the custom field to an external data source's table column. For a lookup relationship field, specify the external table column that contains 18-character Salesforce IDs.
External ID	For each object that can have custom fields, you can set up to seven custom auto-number, email, number, or text fields as external IDs. An external ID field contains record identifiers from a system outside of Salesforce. You can use an external ID field to update or upsert records using the API. When using the API or the Data Import Wizard for custom objects and solutions, you can use this field to prevent duplicates by also marking the field as Unique. Note: Custom fields marked as `Unique` count against an object's limit of seven `External ID` fields. Custom indexing that occurs automatically in the background by Salesforce does *not* count against External ID limits. Not available for external objects. Each external object has an `External ID` standard field. Its values uniquely identify each external object record in your org.
Filter Criteria	The criteria used to select a group of records to calculate the value of a roll-up summary field.
Filtering Disabled	For custom fields on external objects, determines whether the field is available in filters.
Formulas	Enter the formula for the custom formula field or custom summary formula for reports.
Help Text	The text that displays in the field-level help hover text for this field.
Is Name Field	For external object fields of type text, specifies this custom field as the name field for the external object. Not available for text area fields. By default, the External ID standard field is the name field for the external object. If you select this checkbox, make sure that the External Column Name specifies a table column that contains name values. Each external object can have only one name field.

Field	Description
	For internal use only, Salesforce stores the value of the name field from each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.
Label	The name of the custom field as you want it to appear.
Latitude and Longitude Display Notation	For geolocation fields, determines how the latitude and longitude notation appears in the Salesforce interface. **Degrees, Minutes, Seconds** A notation for angular measurement that is based on the number 60: there are 360 degrees to a circle, 60 minutes to a degree, and 60 seconds to a minute. **Decimal** Expresses the value as degrees, and converts the minutes and seconds to a decimal fraction of the degree. Decimal notation doesn't use cardinal points. North and East are positive values; South and West are negative values. For example, the coordinates for San Francisco can be expressed as follows: | | Latitude | Longitude | | --- | --- | --- | | Degrees, Minutes, Seconds | 37° 46' 30" N | 122° 25' 5" W | | Decimals | 37.794016° | –122.395016° | Regardless of the notation you choose to display in the interface, latitude and longitude are stored in Salesforce as decimals.
Length (for text fields)	For text fields, the maximum number of characters that a user can enter in a field (up to 255 characters).
Length (for number, currency, percent fields)	For number, currency, and percent fields, the number of digits you can enter to the left of the decimal point, for example, 123.98 for an entry of 3.
Mask Character	For encrypted text fields, determines the character to use for hidden characters. Available options are * and X.
Mask Type	For encrypted text fields, determines which characters are hidden and the use of dashes in the field. Masked characters are hidden using the character selected in `Mask Character`. Available options are: **Mask All Characters** All characters in the field are hidden. **Last Four Characters Clear** All characters are hidden but the last four display.

Field	Description
	Credit Card Number The first 12 characters are hidden and the last four display. Salesforce automatically inserts a dash after every fourth character. **National Insurance Number** All characters are hidden. Salesforce automatically inserts spaces after each pair of characters if the field contains nine characters. Use this option for UK NINO fields. **Social Security Number** The first five characters are hidden and the last four display. Salesforce automatically inserts a dash after the third and fifth characters. **Social Insurance Number** All characters are hidden but the last three display. Salesforce automatically inserts a dash after the third and sixth characters.
Master Object	The object on the master side of a master-detail relationship used to display the value of a roll-up summary field.
Related List Label	For relationship fields, the title for the related list that displays associated records on the parent record.
Related To	For relationship fields, the name of the associated object.
Required	Makes the field required everywhere in Salesforce. Not available for external objects. You must specify a default value for required campaign member custom fields. Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See Require Field Input to Ensure Data Quality on page 210.
Roll-Up Type	For roll-up summary fields, choose the type of calculation to make: **Type** — **Description** COUNT — Totals the number of related records. SUM — Totals the values in the field you select in the `Field to Aggregate` option. Only number, currency, and percent fields are available. MIN — Displays the lowest value of the field you select in the `Field to Aggregate` option for all directly related records. Only number, currency, percent, date, and date/time fields are available. MAX — Displays the highest value of the field you select in the `Field to Aggregate` option for all directly related records. Only number, currency, percent, date, and date/time fields are available.

Field	Description
Starting Number	For auto-number fields, enter a `Starting Number` that's less than 1 billion. Select `Generate Auto Number for existing records` to automatically number all current records that begin with the starting number that you enter. If deselected, the next record that you enter is assigned the starting number and your older records are blank in this field. For leads, only unconverted leads are assigned a number. When you create records, `Starting Number`'s value increments to store the number that will be assigned to the next auto-number field created. You can't edit `Starting Number` after creating an auto-number field. To edit a `Starting Number` value, change your auto-number field to a text field and then back to an auto-number field. To restart `Starting Number` values for fields on objects from a managed package, uninstall and then reinstall the package. Warning: Be sure that you don't create records with duplicate auto-number values. An auto-number field can contain up to 10 digits and up to 20 extra characters for your prefix or suffix. Note: • You can't retrieve the starting number of an auto-number field through Metadata API. To specify a `Starting Number` while deploying, add a `startingNumber` tag for your field to your `package.xml` file. For example: `<startingNumber>42</startingNumber>` • If you deploy without specifying a `Starting Number` value in your `package.xml` file, the default starting number for standard fields is `0`. The default starting number for custom fields is `1`.
Sharing Setting	For master-detail relationship fields, the `Sharing Setting` attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.
Sorting Disabled	For custom fields on external objects, determines whether the field is sortable.
Summarized Object	The object on the detail side of a master-detail relationship used to provide the values calculated in a roll-up summary field.
Unique	If checked, prevents duplicate field values. For text fields, you can control whether values that are identical except for their case are considered unique. Select `Treat "ABC" and "abc" as duplicate values` to enforce case-insensitive uniqueness, or select `Treat "ABC" and "abc" as different values` to enforce case-sensitive uniqueness.

Field	Description
	Note: Some characters have both single-byte and double-byte versions. For example, all the following characters have single-byte and double-byte versions: “!@#$%^&*(){}[]|:";',.<>?/~`”.  For the purpose of unique field value comparison, the single-byte and double-byte versions of these characters are considered identical.  Note: Custom fields marked as `Unique` count against an object's limit of seven `External ID` fields. Custom indexing that occurs automatically in the background by Salesforce does *not* count against External ID limits.
Values	For picklist fields, a list of available values (up to 255 characters for each value). For picklists, optionally choose to alphabetize the picklist entries. You can also set the first value as the default selection. If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default. For multi-select picklists, enter a list of values, select the sorting options that apply, and then enter how many values you want displayed at a time on edit pages. The number of values determines the box height.

Auto-Number Formatting Examples

Use these examples when setting the display format for auto-number fields.

Format	Displayed Values
{0}	3 66 103
{000}	003 066 103
Sample- {00000}	Sample- 00003 Sample- 00666 Sample- 10023
Value- {00} {MM} {DD} {YY}	Value- 03 12 02 04 Value- 76 03 03 04 Value- 123 11 09 04
PO #{0} {MM}-{DD}-{YY}	PO #12233 12-20-04 PO #25 06-07-04 PO #3 07-07-04

SEE ALSO:

- Create Custom Fields
- Create a Many-to-Many Relationship
- Object Reference for Salesforce and Force.com

Geolocation Custom Field

The geolocation custom field allows you to identify locations by their latitude and longitude and to calculate distances between locations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

You can calculate the distance between two geolocation fields, such as between a warehouse and a store. Or you can calculate the distance between a geolocation field and a pair of latitude and longitude coordinates, such as between a warehouse and 37.794016°, -122.395016°—the location also known as San Francisco. Latitude values must be within `-90` and `90`. Longitude values must be within `-180` and `180`.

Geolocation is a compound field that counts toward your org's limits as three custom fields: one for latitude, one for longitude, and one for internal use. Support for the compound field (geolocation) versus the field's components (latitude and longitude) varies depending on the functionality you're using in Salesforce. For example, you can create list views that show the field and its components, but you can't select the compound geolocation field in Apex. You can run SOQL queries only on a geolocation field's components.

Compound fields, including geolocation fields, have the following limitations.

- Compound fields are read-only. To update field values, modify the individual field components.
- Compound fields are accessible only through the SOAP and REST APIs. The compound versions of fields aren't accessible anywhere in the Salesforce user interface.
- Although compound fields can be queried with the `Location` and `Address` Apex classes, they're editable only as components of the actual field. Read and set geolocation field components by appending "__latitude__s" or "__longitude__s" to the field name, instead of the usual "__c." For example:

  ```
  Double theLatitude = myObject__c.aLocation__latitude__s;
  myObject__c.aLocation__longitude__s = theLongitude;
  ```

  You can't access or set the compound value.
- You can't use compound fields in Visualforce—for example, in an `<apex:outputField>`. To access or update field values, use the individual field components.
- If you select compound fields for export in the Data Loader, they cause error messages. To export values, use individual field components.
- Custom geolocation and location fields on standard addresses aren't supported with email templates.
- You can't use compound fields in lookup filters, except to filter distances that are within or not within given ranges. You can use distance lookup filters only in the Metadata API.
- The only formula functions that you can use with compound fields are `ISBLANK`, `ISCHANGED`, and `ISNULL`. You can't use `BLANKVALUE`, `CASE`, `NULLVALUE`, `PRIORVALUE`, or the equality and comparison operators with compound fields. The equality and comparison operators include `=` and `==` (equal), `<>` and `!=` (not equal), `<` (less than), `>` (greater than), `<=` (less than or equal), `>=` (greater than or equal), `&&` (AND), and `||` (OR).

Other limitations of geolocation fields include:

- Geolocation fields aren't supported in custom settings.
- Geolocation fields aren't available in dashboards or Schema Builder.
- Geolocation fields are available in Visual Workflow and in formula-based workflow and approvals, but they can't be used in filter-based workflow updates and approvals.
- `DISTANCE` formulas are supported in:
  - Entry criteria for workflow rules and approval processes
  - Field update actions in workflow rules and approval processes

  - Custom validation rules
  - Lookup filters (in the Metadata API only)

- Geolocation fields and latitude and longitude on standard addresses aren't supported in Salesforce to Salesforce.
- In Developer, Professional, Enterprise, Unlimited, and Performance editions, Salesforce can automatically add or update geolocation fields for Account, Contact, Lead, and WorkOrder records. To use this feature, your administrator must enable the geo clean rule for each object. For all other objects and editions, set values for latitude and longitude by using SOQL, Workbench, SOAP or REST API, or a geocoding service. You can then use address fields as locatable values. To find geocoding services, search the AppExchange.
- Geolocation fields are supported in SOQL with the following limitations.
  - `DISTANCE` and `GEOLOCATION` are supported in `WHERE` and `ORDER BY` clauses in SOQL, but not in `GROUP BY`. `DISTANCE` is supported in `SELECT` clauses.
  - `DISTANCE` supports only the logical operators > and <, returning values within (<) or beyond (>) a specified radius.
  - When using the `GEOLOCATION` function in SOQL queries, the geolocation field must precede the latitude and longitude coordinates. For example, `DISTANCE(warehouse_location__c, GEOLOCATION(37.775,-122.418), 'km')` works but `DISTANCE(GEOLOCATION(37.775,-122.418), warehouse_location__c, 'km')` doesn't work.
  - Apex bind variables aren't supported for the units parameter in `DISTANCE` or `GEOLOCATION` functions. This query doesn't work.

```
String units = 'mi';
List<Account> accountList =
    [SELECT ID, Name, BillingLatitude, BillingLongitude
     FROM Account
     WHERE DISTANCE(My_Location_Field__c, GEOLOCATION(10,10), :units) < 10];
```

For more information and examples, see the *Force.com SOQL and SOSL Reference*.

SEE ALSO:

Custom Field Types

Custom Field Attributes

Manage Fields for a Specific Object

1. From the object management settings for the object whose fields you want to view, go to Fields.
2. Click the field label.
3. To modify a custom field, add custom help text, or change the data type, click **Edit**.
4. If a custom field exists in a Managed - Released package, click **Delete** to delete the custom field component from future installations.
5. To set users' access to the field, click **Set Field-Level Security**. This option is available depending on the edition that you have.
6. To view who can access the field based on permissions and record type, click **View Field Accessibility**. This option is available depending on the edition that you have.
7. If the custom field is a dependent picklist, click **Change** next to the controlling field to edit the dependency rules.
8. To change `External ID`, `Required`, or other attributes under the General Options section, see Custom Field Attributes.
9. To restore the field and its data, click **Undelete**. This option is available only if the field has been deleted but not permanently erased. The field's behavior may be different after restoring it. To restore the field completely, see Manage Deleted Custom Fields.

Note: If your organization uses person accounts, the Account Fields page lists both person account and business account fields.

Person accounts use a combination of account and contact fields. The following contact fields are available for person accounts, but not for business accounts.

- Fields in the Account Standard Fields list that display with a person account icon.
- Fields in the Contact Custom Fields & Relationships list.

SEE ALSO:

Custom Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard fields are not available in **Database.com**

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Edit Custom Fields

1. From the management settings for the field's object, go to Fields.
2. Click **Edit** next to the field's name.
3. Modify the field attributes. The attributes differ depending on the field type.

   If you're editing a picklist, you can change its definition and its values. For picklist settings, see Add or Edit Picklist Values on page 149.

   To change the type of this custom field, see Change the Custom Field Type on page 164.
4. Optionally, define custom help text for the field.
5. For lookup and master-detail relationship fields, optionally define a lookup filter.
6. For formula fields, click **Next** to modify the formula.
7. In Enterprise, Unlimited, Performance, and Developer Editions, click **Next** to set the field-level security for the field.

Note:

- Editing fields may require changing a large number of records at once. To process these changes efficiently, your request may be queued and you may receive an email notification when the process has completed.
- To customize the way a custom object's related list appears on a parent record's detail page, edit the parent record's page layout. For example, if you want to edit which fields appear on a custom object's related list on accounts, you would edit the account page layout.
- You cannot change the `Field Name` if a custom field is referenced in Apex.
- When editing fields for accounts, opportunities, cases, contacts, or custom objects, check for any criteria-based sharing rules that use the field in the rules. A field change may affect which records are shared.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To create or change fields:
- "Customize Application"

SEE ALSO:

Define Default Field Values

Find Object Management Settings

Delete Fields

1. From the management settings for the field's object, go to Fields.
2. Click **Del** next to the name of the field.
3. When prompted, select the `Yes, I want to delete the custom field` checkbox to confirm, and click **Delete**.

   Deleted custom fields and their data are stored until your org permanently deletes them or 15 days has elapsed, whichever happens first. Until that time, you can restore the field and its data. For information on restoring deleted custom fields and relationships, see Manage Deleted Custom Fields on page 144.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To delete custom fields:
- "Customize Application"

Note:

- Before deleting a custom field, consider where it's referenced. You can't delete a custom field that's referenced elsewhere. For example, you can't delete a custom field that's referenced by a field update or Apex.
- You can't delete a field if that field is being updated by a background job, such as an update to a roll-up summary field. Wait until the background job finishes, and try again.
- When you delete a custom field, all of the field history data is deleted and changes are no longer tracked.
- A background process periodically runs that cleans up metadata associated with deleted custom fields. This process will affect the `Last Modified Date` and `Last Modified By` fields on page layouts, record types, and custom objects.

SEE ALSO:

Find Object Management Settings

Manage Deleted Custom Fields

Deleted custom fields and their data are stored until your org permanently deletes them or 15 days has elapsed, whichever happens first. Until that time, you can restore the field and its data.

However, the field still counts against the maximum number of custom fields allowed in your organization. A deleted field also counts against any applicable limit for its field type. For example, a deleted custom roll-up summary field counts against the maximum number of roll-up summary fields for your Salesforce edition.

1. From the management settings for the field's object, go to Fields.
2. Click **Deleted Fields** at the bottom of the list of custom fields and relationships. The number in parentheses indicates the total number of deleted custom fields for this object. This link only displays when you have a deleted custom field.
3. Use the list of deleted fields to perform the following actions:
   - To view details about a field, click the field label.
   - To permanently remove the custom field and its data, click **Erase**.
   - To restore the field and its data, click **Undelete**. Some attributes of deleted fields are not restored automatically. To restore these attributes manually:
     a. Add the field to any page layouts that changed during the time the custom field was deleted. If reports and page layouts were not edited, the restored field remains on them.
     b. Make the field unique if necessary. Salesforce automatically removes the unique attribute from any deleted custom field.
     c. Make the field required if necessary. Salesforce automatically removes the required attribute for any deleted custom field.
     d. Add the custom field to any appropriate Force.com AppExchange packages. Salesforce automatically removes deleted custom fields from packages that contain them.
     e. Convert any lookup relationships to master-detail relationships if necessary. Salesforce converts all relationships to lookup relationships when they are deleted. To convert a lookup relationship to a master-detail relationship, populate all the applicable records with the appropriate data.
     f. Redefine any field dependencies that Salesforce removed when the field was deleted.
     g. Edit and save any formula fields. Saving prompts a syntax check; if necessary, resolve errors.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Page Layouts and Lead Fields are not available in **Database.com**

USER PERMISSIONS

To restore deleted custom fields and relationships:
- "Customize Application"

To permanently delete custom fields or relationships:
- "Customize Application"

h. Set up field history tracking if necessary. If the list of fields enabled for history tracking was modified during the deletion process, the restored field is no longer set up to track field history.

Notes on Restored Custom Fields

- When deleted, the following characters are appended to the end of a custom field's developer name unless a deleted field already has that developer name: "_del". These characters remain when you restore the custom field.
- Formula fields are restored in a disabled state, which means they do not contain updated data until you edit and save them. While a formula field is disabled, "#Error!" displays in place of the formula value.
- Restored fields do not display in search results immediately after you restore them. It can take a short time before the restored custom field and its data are available in search results.
- Lead fields that are mapped to account, contact, or opportunity fields for lead conversion are still mapped accordingly when restored.
- Auto number fields continue to increment after they are deleted and contain the correct values when restored.
- Field history data for the deleted custom field is restored.

SEE ALSO:

Find Object Management Settings

Additional Custom Field Options

Changing Page Layouts
To change the location of a new custom field, edit the page layout for the appropriate tab.

Using Record Types
If your organization uses record types, edit the record type to modify which picklist values are visible for the record type.

Tracking Custom Field History
You can select which custom fields to track on the History related list of custom objects and most standard objects. All entries include the date, time, nature of the change, and who made the change. History data does not count against your organization's storage limit.

Using the Translation Workbench
If your organization uses the Translation Workbench, notify your translators that new fields need translations.

Activity Custom Fields
Activity custom fields can apply only to tasks or only to events, or to both tasks and events. For example, you can create one `Time Spent` field and then add it to both the event page layout and the task page layout.

Mapping Custom Lead Fields
For lead custom fields, you can click **Map Lead Fields** to specify which custom lead fields to map to custom account, contact, and opportunity fields during a lead conversion.

Rich Text Area Field Tips

Use rich area text fields to improve the appearance of text, including adding images and hyperlinks.

Rich text area fields use the Salesforce HTML editor to format content.

Note: The upgraded editor doesn't support Internet Explorer version 7 or version 8 in compatibility mode. If you are using these browsers, you will use the older editor.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To create or change custom fields:
- "Customize Application"

Implementation Tips

- Specify how large the editor box should be for a rich text field by configuring the "Number of lines displayed" property in the field's setup.
- When you view or print content, Salesforce preserves the formatted version of the HTML in rich text area fields.
- Searches of content that contains rich text area fields ignore images and tags.
- Deleting a rich text area field moves it to the Deleted Fields section on the custom object or Salesforce Knowledge article types.
- You can only convert rich text area fields into long text area fields. Any images are deleted the next time the long text area field is saved. After converting, markup is hidden in the long text area field but it is not removed from the record until you save the record. That way, the markup can be restored if you change your mind.
- JavaScript or CSS is treated as text. For example, if you're creating an Idea through the API, any JavaScript or CSS code is removed without warning. Salesforce supports a limited number of approved HTML tags.
- When a rich text area field is used in a formula, the HTML tags are stripped out before the formula is run.
- Rich text area fields can be filtered and summarized in reports, but HTML tags and special formatting aren't included in report results. For example, `<b>some</b> <i>text</i>` becomes "some text" instead of some text or **some** *text*.
- The text part of rich text area fields counts towards data storage for the object that contains the field.
- You can copy and paste text from external sources such as Microsoft® Word, but you may have to reapply formatting.
- You can use a rich text area field in a mail merge, but the HTML tags are included as text in the resulting document, and images aren't merged.

Images in Rich Text Area Fields

- The maximum size of an image that can be uploaded in a rich text area field is 1 MB. Only `.gif`, `.jpg`, and `.png` file types are supported.
- To upload many images, use API version 20 or later. For more information, see the *Force.com SOAP API Developer's Guide*.

  Note: When you upload images via the API, the `alt` attribute isn't populated unless you specified it separately.

- Images uploaded within a rich text area field are extracted in the organization's weekly export and included in the exported data.
- Images within the rich text area fields count towards file storage for the object that contains the field.
- You can't add a hyperlink to an image.
- You can't upload an image to a rich text area using the *`file://`* protocol in the **URL** field. Instead, use *`http:`*, *`https:`*, *`data:`*, *`//`*, */*, or a relative URL.

SEE ALSO:

Create Custom Fields

Rich Text Area Field Limitations

Keep these limitations in mind when working with rich text area fields.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

- Salesforce supports up to 131,072 characters for each rich text area field, including the HTML tags. If desired, you can set a lower limit.
- An object can contain unlimited rich text area and long text area fields, although your edition's limit for total custom fields allowed on an object, regardless of field type, applies. Each object can contain 1,600,000 characters across long text area and rich text area fields. When you create a long text area or rich text area field, you set a character limit for the field—the maximum length of the text that can be entered. The default character limit for long text area and rich text area fields is 32,768 (32 KB). The maximum character limit for long text area and rich text area fields is 131,072 (128 KB). The minimum character limit is 256.
- You can't paste special characters, such as bullets or curly quotes, into a rich text field from another application. It's best to type or paste in plain text and use the rich text editor to format it.
- You can't disable specific rich text area features. For example, you can't disable support for hyperlinks or images on certain fields.
- HTML code isn't supported in the Salesforce HTML editor. HTML code is treated as text.
- Rich text area fields aren't available in self service portals.
- Rich text area fields aren't supported for external objects.
- Only the first 254 characters in a rich text area or a long text area are supported with the "contains" operator in a report filter.
- Only the first 254 characters in a rich text area or a long text area are displayed in a report. If you download the report, the entire field is available.
- You can't upload an image to a rich text area using the `file://` protocol in the **URL** field. Instead, use `http:`, `https:`, `data:`, `//`, `/`, or a relative URL.

Classic Encryption for Custom Fields

Restrict other Salesforce users from seeing custom text fields you want to keep private. Only users with the permission "View Encrypted Data" can see data in encrypted custom text fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, **Unlimited**, and **Database.com** Editions

Note: This information is about Classic Encryption and not Shield Platform Encryption.

Before you begin working with encrypted custom fields, review these implementation notes, restrictions, and best practices.

Implementation Notes

- Encrypted fields are encrypted with 128-bit master keys and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact Salesforce.
- You can use encrypted fields in email templates but the value is always masked regardless of whether you have the "View Encrypted Data" permission.
- If you have created encrypted custom fields, make sure that your organization has "Require secure connections (HTTPS)" enabled.
- If you have the "View Encrypted Data" permission and you grant login access to another user, the user can see encrypted fields in plain text.
- Only users with the "View Encrypted Data" permission can clone the value of an encrypted field when cloning that record.
- Only the `<apex:outputField>` component supports presenting encrypted fields in Visualforce pages.

Restrictions

Encrypted text fields:

- Cannot be unique, have an external ID, or have default values.
- For leads are not available for mapping to other objects.
- Are limited to 175 characters because of the encryption algorithm.
- Are not available for use in filters such as list views, reports, roll-up summary fields, and rule filters.
- Cannot be used to define report criteria, but they can be included in report results.
- Are not searchable, but they can be included in search results.
- Are not available for: Salesforce Classic Mobile, Connect Offline, Salesforce for Outlook, lead conversion, workflow rule criteria or formulas, formula fields, outbound messages, default values, and Web-to-Lead and Web-to-Case forms.

Best Practices

- Encrypted fields are editable regardless of whether the user has the "View Encrypted Data" permission. Use validation rules, field-level security settings, or page layout settings to prevent users from editing encrypted fields.
- You can still validate the values of encrypted fields using validation rules or Apex. Both work regardless of whether the user has the "View Encrypted Data" permission.
- Encrypted field data is not always masked in the debug log. Encrypted field data is masked if the Apex request originates from an Apex Web service, a trigger, a workflow, an inline Visualforce page (a page embedded in a page layout), or a Visualforce email template. In other cases, encrypted field data isn't masked in the debug log, like for example when running Apex from the Developer Console.
- Existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type. To encrypt the values of an existing (unencrypted) field, export the data, create an encrypted custom field to store that data, and import that data into the new encrypted field.
- `Mask Type` is not an input mask that ensures the data matches the `Mask Type`. Use validation rules to ensure that the data entered matches the mask type selected.
- Use encrypted custom fields only when government regulations require it because they involve more processing and have search-related limitations.

Note: This page is about Classic Encryption, not Shield Platform Encryption. What's the difference?

SEE ALSO:

Create Custom Fields

Add or Edit Picklist Values

Add or edit values in a custom picklist from the fields area of an object. If the picklist uses a global picklist value set, you can change its values only by editing the global value set. Your changes affect all picklists that inherit their values from that global value set.

Note: Changes to picklist values are logged in Setup Audit Trail. To view the audit history: From Setup, enter *View Setup Audit Trail* in the Quick Find box, then select **View Setup Audit Trail**.

1. Navigate to the fields area for your object.
2. In the Custom Fields & Relationships related list, click the name of the picklist field to update.
3. In the Values section, click **Edit** next to a value.
4. Change the value's name, and optionally make the value the default for the master picklist.
5. Assign a color to use in charts by clicking the button and selecting how to assign colors to values.
   - **Assign fixed colors to all values** assigns a fixed color to each value from the standard set of chart colors. The Chart Colors column shows the assigned colors. For example, if you want Closed Lost values always to be red in charts grouped by Opportunity Stage, assign red to that picklist value.
   - **Assign colors to values dynamically** assigns colors when a chart is generated. For example, if you need only certain picklist values to have fixed colors in charts, manually assign colors to those values and leave the rest to be assigned dynamically.

   Note: Chart colors aren't available for multi-select picklists, currency picklists, or Task Subject, Event Subject, Knowledge Validation Status, and Opportunity Competitor picklists.

   Tip:
   - If you use record types, changing the default value of the master picklist doesn't affect the default value of the picklist for a record type.
   - For Ideas, setting the default value of the Categories or Status picklists doesn't affect the default value on the Ideas pages.
   - If you change the label for a picklist value that's used as a filter criterion, the picklist value is automatically removed from the filter criteria. For example, if your report contains a filter where *Lead Source equals Email or Web* and you change the picklist value Web to Referral, your report filter changes to *Lead Source equals Email*. If the changed picklist value was the only value specified for a particular filter, it continues to show up in your filters, but an error appears.
6. Click **Save**.
7. To make the picklist required (if it's not already), click **Edit** at the top of the detail page, and check **Always require a value in this field in order to save a record**.
8. To change the picklist from unrestricted to restricted or vice-versa, adjust the **Restrict picklist to the values defined in the value set** setting.
9. To open an easy-to-print list of your picklist values, click **Printable View**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To change picklists:
- "Customize Application"

Note: If your org uses the Translation Workbench, notify your translators that the translations can be outdated when you change picklist values.

SEE ALSO:

- Manage Inactive Picklist Values
- Manage Inactive Picklist Values
- Replace Picklist Values
- Picklist Limitations

Picklists with Additional Information

These standard picklist fields have additional information that you can edit.

Picklist	Description
`Partner Role` (for accounts)	Roles of account partners, for example, Consultant, Supplier. These options are available when you add an account to the Partners related list of an account or opportunity. To edit, from Setup, enter *`Partner Roles`* in the `Quick Find` box, then select **Partner Roles**. Enter the name of the partner role in the "Role" column. In the "Reverse Role" column, enter the corresponding reverse partner role. Assigning a partner role to an account creates a reverse partner relationship so that both accounts list the other as a partner. Each role and reverse role value can have up to 40 characters.
`Priority` (for cases)	Urgency of case, for example, Low, High. If you delete a value, you have the option to map the deleted value to another existing value in all your org's cases. Each picklist value can have up to 40 characters.
`Status` (for campaign members)	State of a campaign member, for example, Sent or Responded. If you delete a Status value, you have the option to map the deleted value to another existing value. The new replacement value is automatically added to the member status for campaigns that contained the deleted value. If the deleted value is the default member status for a campaign, the new replacement value becomes the default status for that campaign.
`Status` (for cases)	State of case, for example, New, On Hold. If you delete a value, you have the option to map the deleted value to another existing value in all your org's cases. Each picklist value can have up to 40 characters.

Picklist	Description
`Status` (for contracts)	State of the contract in the contract business process. You can add values to this picklist and organize each value into one of several categories, for example, "Draft", "In Approval Process", or "Activated". Then sort your contracts using these categories in reports and views.
`Contact Role` (for contracts)	Role of a contact on a contract, for example, Business User, Decision Maker. These options are available when you add a contact to the Contact Roles related list of a contract. To edit, from Setup, enter *`Contact Roles`* in the `Quick Find` box, then select **Contact Roles on Contracts**. Each picklist value can have up to 40 characters.
`Lead Status` (for leads)	State of the lead, for example, Open, Qualified. Select one value as the "Default Status" assigned to all new leads created manually, via imports, or via your website. Select one or more values as the "Converted Status" assigned to converted leads. When you convert qualified leads into an account, contact, and opportunity, you can select one of the "Converted" statuses to assign to the lead. Leads with a "Converted" status type are no longer available in the Leads tab, although you can include them in reports. If you delete a value, you have the option to map the deleted value to another existing value in all your org's leads. Each value can have up to 20 characters.
`Contact Role` (for opportunities)	Role of a contact for an opportunity, for example, Business User, Decision Maker. These options are available when you add a contact to the Contact Roles related list of an opportunity. To edit, from Setup, enter *`Contact Roles`* in the `Quick Find` box, then select **Contact Roles on Opportunities**. Each picklist value can have up to 40 characters.
`Stage` (for opportunities)	Sales process stages, for example, Prospect, Proposal. This picklist also affects the `Type` and `Forecast Category` values of an opportunity. Specifically, changing the `Type` or `Forecast Category` for a `Stage` picklist value updates all opportunities that have that stage value. To edit, from the management settings for opportunities, go to Fields, and then click **Edit** next to Stage. To deactivate an active stage, click **Del** next to the stage. On the mapping page, don't replace the stage with another existing value; just click **Save**. The stage now appears in the Inactive Stage Picklist Values related list. The stage is no longer in use but can exist in older opportunity records.

Picklist	Description
`Status` (for solutions)	Status of a solution, for example, Draft, Reviewed. Mark one or more values as "Reviewed". When users solve cases using solutions, they can view which solutions have been reviewed and which have not. Each picklist value can have up to 40 characters.
`Priority` (for tasks)	Importance of the task, for example, High, Normal, Low. Set one value as the default priority of all new tasks, and one value as the highest priority. If you delete a value, you have the option to map the deleted value to another existing value in all your org's tasks. Each picklist value can have up to 20 characters.
`Status` (for tasks)	State of a task, for example, Not Started, Completed. Choose at least one value as the "Closed" status and one value as the "Default" status for all new tasks. If you delete a value, you have the option to map the deleted value to another existing value in all your org's tasks. Each picklist value can have up to 40 characters.
`Task Type` (for tasks)	`Send Email Default` specifies the default task type assigned when the task is sending email or mass email, and when tasks are created via inbound email, such as Email to Salesforce. `Default` specifies the default picklist value when creating tasks. To edit, from the management settings for tasks, go to Fields, and then click **Edit** next to the picklist value that you want to specify as the default.

SEE ALSO:

Picklist Limitations

Picklist Limitations

The maximum number of characters you can have in a picklist depends on the type of picklist. Each value in a picklist includes a line break and a return character that aren't visible. These two additional characters per value are counted as part of the character limit for each value.

Additional limits apply to both standard and custom picklists.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Standard Picklists are not available in **Database.com**

USER PERMISSIONS

To change picklists:
- "Customize Application"

Additional Limits for Standard Picklists

For standard picklists, each value can have up to 255 characters, not including line breaks and returns. This applies to single-select and multi-select picklists.

For standard picklists in orgs that use record types or the Translation Workbench, you can have unlimited values with these exceptions for special picklists.

Picklist Field	Maximum Number of Values
`Lead Status`	100
`Task Status`	100
`Task Priority`	50
`Case Status`	100
`Case Priority`	50
`Opportunity Stage`	100

Limits for Custom Picklists

Custom single-select picklists can have:

- Up to 1,000 values
- Up to 255 characters per value

Note: These limits apply to custom restricted *and* unrestricted picklists.

Custom multi-select picklists can have:

- Up to 500 values
- Up to 255 characters per value

Users can select up to 100 values at a time on a record.

You can have up to 500 global picklists (or picklist value sets) in an org. Each global value set can contain up to 500 values. These limits don't apply to custom picklists that use global picklist value sets.

Functional Limitations for Custom Picklists

- You can make a custom picklist field into a restricted picklist only if the picklist has fewer than 1,000 values (or entries). A restricted picklist's values are limited to only those values defined by a Salesforce admin. This prevents users from loading redundant or erroneous values through the API.
- Global picklist value sets are always restricted picklists. This is a good thing, as it preserves data integrity. (Global value sets are shared across objects. Reuse the value set for any custom picklist field.)

- For custom picklist fields that use a global picklist value set, you can change from a single-select to multi-select picklist and vice versa. However, you can't change the picklist to a different field type such as checkbox, currency, or text.

Replace Picklist Values

As your business needs change, replace picklist values with more relevant alternatives. Replacing a value globally replaces that field value on all existing records.

For example, let's say you have a status picklist with three different closed values, Closed-red, Closed-yellow, and Closed-green, and you want to simplify those to just one value. Replace them with the new value Closed.

Important: If you replace a parent value in a controlling picklist, the picklist dependency on that value is lost. After replacing the parent value, re-create the dependency using the new parent value.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To replace picklist values:
- "Customize Application"

1. If necessary, create the replacement value in the picklist edit page. See Add or Edit Picklist Values.
2. Navigate to the picklist.
   - For a global picklist value set: From Setup, enter *picklist* in the `Quick Find` box, then select **Picklist Value Sets**.
   - For a picklist on an object, go to the fields area of the object. For example, for an Account picklist: From Setup, enter *Account* in the `Quick Find` box, then select **Fields** under Accounts.
3. Start the picklist value replace process.
   - For a global picklist value set: Go to the Global Value Set Detail page by clicking the picklist name. In the Values section, click **Replace**.
   - For all other picklists: Click **Replace** next to the picklist name.
4. Enter the value to change, and select a new replacement value.

   Note: Replacing an existing picklist value also changes the `Modified By` date and time for the record.
5. To use the new value in records where this field is currently empty, select `Replace all blank values`.
6. To update all occurrences of the value in your org's records with the new value, click **Replace**. Occurrences in the Recycle Bin are also updated.

Your replace job is queued. To check the job's status, from Setup, enter *Background Jobs* in the `Quick Find` box, then select **Background Jobs**. You receive an email when the job is complete.

Note: If you replace the Stage picklist for opportunities, the `Probability`, `Forecast Category`, and `Expected Revenue` fields are also updated with the corresponding values.

SEE ALSO:

Create a Global Picklist Value Set

Replace Picklist Values

Manage Inactive Picklist Values

Sort Picklists

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To sort picklists:
- "Customize Application"

1. From the management settings for the picklist field's object, go to Fields.

   Note:
   - For custom task or event fields, go to the object management settings for Activities.
   - For standard task fields, go to the object management settings for Tasks. For standard event fields, go to the object management settings for Events.
   - For Knowledge validation status picklists, from Setup, enter *Validation Statuses* in the Quick Find box, then select **Validation Statuses**.

2. Click the name of the picklist to update.
3. Click **Reorder**.
4. Use the arrows to arrange the field in the proper sequence.
5. Select a default value if desired.
6. To alphabetize the entries for users on edit pages, check **Sort values alphabetically...**. The entries always appear in alphabetical order, regardless of the user's language.
7. Save your changes.

Note: On record edit and detail pages and in reports, picklist and multi-select picklist fields can include inactive values. These inactive values are sorted last, unless you've chosen alphabetical sorting. In that case, all values are sorted alphabetically.

SEE ALSO:

Find Object Management Settings

Manage Inactive Picklist Values

Deactivate, reactivate, or remove a value from a restricted or unrestricted custom picklist. If it's a global picklist value set, these actions simultaneously update all custom picklists that inherit its value set.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS

To create or change custom fields:
- "Customize Application"

Note: Only a Salesforce admin can modify the values of a restricted picklist. Users can't add unapproved values, even through the API.

1. Navigate to the picklist you want to modify.
   - For a picklist on an object, go to the fields area of the object.
   - For a global picklist: From Setup, enter *picklist* in the Quick Find box, then select **Picklist Value Sets**.
2. Go to the picklist's detail page by clicking the picklist's name.
3. In the Values section, modify the picklist value.
   - To deactivate a value, which removes it from the picklist but keeps it on existing records: Click **Deactivate** next to the value's name. The value moves to the Inactive Values section. If you need the value again later, click **Activate** next to its name.
   - To remove a value from the picklist and all records that use it: Click **Del** next to the value's name and select **Replace value on records with blank value**. Then click **Save**.

- To remove a value from a *global* picklist and any custom picklists that share its value set: Click **Del** next to the value's name and then click **OK**. If any custom picklists use this global picklist value set, you're prompted to replace the value with either a blank value or an existing, active value. If no custom picklists use the value set, the value is deleted from the global picklist.

  Note: If a global picklist value is used in historical trending, it's deactivated but not deleted.

Deleting a value in a global picklist value set happens immediately. Delete jobs on non-global, restricted picklists go to the background jobs queue. When the job completes, your restricted picklist is updated and you're notified by email.

Changes to picklist values are logged in Setup Audit Trail. To view the audit history: From Setup, enter `View Setup Audit Trail` in the `Quick Find` box, then select **View Setup Audit Trail**.

Define Dependent Picklists

Tip: If your org uses record types, make sure that your controlling and dependent picklist values are available in the appropriate record types before defining a dependent picklist.

1. From the management settings for the object you want to add a field to, go to Fields.

   Custom task and event fields are accessible from the object management settings for Activities.
2. Click **Field Dependencies**.
3. Click **New**.
4. Choose a controlling field and dependent field.

   Note: Some picklist and checkbox fields may not be available as controlling fields.
5. Click **Continue**.
6. Use the field dependency matrix to specify the dependent picklist values that are available when a user selects each controlling field value.
7. Optionally, click **Preview** to test your selections. If your organization uses record types, choose a record type to test how it affects your controlling and dependent picklist values. The record type controls what values are available in the controlling field. The record type and the controlling field together determine what values are available in the dependent picklist. For example, a dependent value is only available if it is available in the selected record type and the selected controlling value.

   Note: The `Filter by Record Type` option doesn't appear in the Preview window for activity custom fields.
8. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To define dependent picklists:
- "Customize Application"

SEE ALSO:

Dependent Picklist Considerations

Edit Dependent Picklists

Delete Picklist Dependencies

Dependent Picklists

Use dependent picklists to help your users enter accurate and consistent data. A *dependent picklist* is a custom or multi-select picklist for which the valid values depend on the value of another field, called the *controlling field*. Controlling fields can be any picklist (with at least one and fewer than 300 values) or checkbox field on the same record.

For example, you can define a `Reason` custom picklist on opportunities and make its valid values depend on the value of the `Stage` picklist as follows:

- If `Stage` is *`Closed Won`*, the valid values for `Reason` are `Superior features` or `Lower price`.
- If `Stage` is *`Closed Lost`*, the valid values for `Reason` are `Inferior features`, `Higher price`, or `Company viability`.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

SEE ALSO:

Define Dependent Picklists

Dependent Picklist Considerations

Using the Field Dependency Matrix

The field dependency matrix lets you specify the dependent picklist values that are available when a user selects each controlling field value. The top row of the matrix contains the controlling field values, while the columns list the dependent field values.

Use this matrix to include or exclude values. Included values are available in the dependent picklist when a value in the controlling field is selected. Excluded fields are not available in the dependent picklist for the selected controlling field value.

To include or exclude values:

- Double-click values to include them. Included values are indicated with highlighting. Double-click any highlighted values to exclude them.
- Click a value and use SHIFT+click on another value to select a range of adjacent values. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
- Click a value and use CTRL+click to select multiple values. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
- Click a column header to select all the values in that column. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.

To change the values in your view:

- Click **View All** to view all available values at once.
- Click **Go To** and choose a controlling value to view all the dependent values in that column.
- Click **Previous** or **Next** to view the values in columns that are on the previous or next page.
- Click **View sets of 5** to view 5 columns at a time.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To define picklist dependencies:

- "Customize Application"

Edit Dependent Picklists

1. From the management settings for the picklist's object, go to Fields.
2. Click **Field Dependencies**.
3. Click **Edit** next to the field dependency relationship you want to change.
4. Use the field dependency matrix to specify the dependent picklist values that are available when a user selects each controlling field value.
5. Optionally, click **Preview** to test your selections.
6. Save your changes.

SEE ALSO:

Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To edit field dependencies:
- "Customize Application"

Delete Picklist Dependencies

If you no longer want the values of a dependent picklist to depend on a controlling field, delete its dependency. Deleting the dependency removes the logic that defines how the values of the picklist depend on the controlling field, but doesn't delete the fields or affect their data.

1. From the management settings for the picklist's object, go to Fields.
2. Click **Field Dependencies**.
3. Click **Del** next to the field dependency relationship you want to delete.
4. Click **OK** to confirm.

SEE ALSO:

Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To delete picklist dependencies:
- "Customize Application"

Dependent Picklist Considerations

Consider the following when defining dependent picklists:

Checkboxes
Checkbox fields can be controlling fields but not dependent fields.

Converting fields
When you convert existing fields to dependent picklists or controlling fields, it doesn't affect the existing values in your records. After conversion, the dependency rules apply to new records and to any changes to existing records.

Default values
You can set default values for controlling fields but not for dependent picklists.

Field-level security
Field-level security settings for a controlling field and dependent picklist are independent. Remember to hide a controlling field whenever its correlating dependent picklist is hidden.

Import
The Data Import Wizard doesn't consider field dependencies. You can import any value into a dependent picklist regardless of the value imported for a controlling field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Page Layouts and Leads are not available in **Database.com**

USER PERMISSIONS

To define and edit dependent picklists:
- "Customize Application"

Lead conversion

If you create a dependency for lead fields that map to account, contact, and opportunity fields for lead conversion, create the same dependency on the account, contact, or opportunity.

Dependent picklists and controlling lead fields can be mapped to account, contact, or opportunity fields with different dependency rules.

Multi-select picklists

Multi-select picklists can be dependent picklists but not controlling fields.

Standard versus custom picklists

Custom picklist fields can be either controlling or dependent fields.

Standard picklist fields can be controlling fields but not dependent fields.

Picklist limitations

A controlling field can have up to 300 values. If a field is both a controlling field and dependent picklist, it can't contain more than 300 values.

The following fields are not available as controlling fields.

Activity Fields

`Call Type`

`Create recurring series of events`

`Show Time As`

`Subject`

`Task`

`Type`

Contact Fields

`Salutation`

`Contact Currency`

Custom Object Fields

`Currency`

Lead Fields

`Converted`

`Unread By Owner`

Dependency limitations

Before defining a dependency, make sure that your picklist has at least one value. Standard fields like `Product Family` do not contain values until you add them.

If a standard controlling field relies on functionality that your organization decides to disable, the dependency rules for the picklist go away. For example, if your organization disables the Self-Service portal and the `Closed by Self-Service User` is a controlling field, its dependent picklist displays all available values.

If you replace a parent value in a controlling picklist, the picklist dependency is lost. After replacing the parent value, re-create the dependency using the new parent value.

Connect Offline

While controlling fields and dependent picklists are available in Connect Offline, the logic between them is not.

Page layouts

Page layouts that contain dependent picklists must also contain their controlling picklist fields, or the dependent picklist values don't display.

For visually impaired users, make sure that the dependent picklist is lower on the page layout than its controlling field.

If a dependent picklist is required and no values are available for it based on the controlling field value, users can save the record without entering a value. The record is saved with no value for that field.

Record types

You can define or change the record type for your dependent picklist only within the Preview dialog when creating or editing the field dependency values. The record type controls what values are available in the controlling field. The record type and the controlling field together determine what values are available in the dependent picklist. For example, a dependent value is only available if it is available in the selected record type and the selected controlling value.

Using the HTML Editor

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom fields:

- "Customize Application"

To improve the appearance of text in custom fields, use the Salesforce HTML editor to format custom fields that were defined with the rich text area custom field type. The Salesforce HTML editor provides a WYSIWYG interface, allowing you to perform basic formatting operations. You can:

- Format text as bold, italicized, or underlined
- Create bulleted and numbered lists
- Change paragraph indentation
- Insert hyperlinks to Web pages
- Insert an image

Note: The upgraded editor doesn't support Internet Explorer version 7 or version 8 in compatibility mode. If you are using these browsers, you will use the older editor.

Some features, such as Salesforce Knowledge or Ideas, let administrators enable extra functions like the ability to embed multimedia content, use drop-down lists, or select colors for text or backgrounds. Make sure that your browser security settings allow you to access these functions. For example, if browser security is set to high in Microsoft® Internet Explorer® 7, `<iframe>` content does not appear in the frame, drop-down lists are unavailable, and color cannot be selected for backgrounds or text.

Common scenarios for using the HTML editor include:

- Pasting a code sample into a Chatter Answers or Ideas post
- Adding a video to a knowledge article or an Idea Themes post
- Formatting resumes submitted by job candidates to create more readable and professional-looking versions for your users
- Emphasizing key points in a descriptions or comments by setting various font options, such as bold typeface and a larger font size
- Itemizing steps in a numbered or bulleted list to help users better understand a procedure
- Supporting your company's brand identity by specifying the URL of a logo on your website
- Including a link to a related and accessible page of your company's website

Notes on Using the Salesforce HTML Editor

When you use the editor, note the following:

- Because the Salesforce HTML editor provides a WYSIWYG interface only, you don't edit HTML tags.
- If you copy content from a Web page and paste it into the editor's window, Salesforce automatically removes unsupported tags and preserves the text that was enclosed in unsupported tags as plain text. Salesforce does not notify users when it removes unsupported or potentially malicious HTML.
- The maximum number of characters you can enter in the Salesforce HTML editor window is equal to the field length specified when creating or editing the field. The default is 32,768 characters.
- When clicked, hyperlinks open in a new browser window. The Salesforce HTML editor supports HTTP, HTTPS, and mailto hyperlinks.
- When a record is saved, the Salesforce HTML editor does not validate hyperlinks to Web pages. Be sure to confirm that you are specifying a URL that Salesforce can access.
- To insert an image, click  and either select:
  - **Web Address** tab and enter the URL of the image.
  - **Upload Image** tab and select an image from your localhost. You can upload only JPEG, PNG or GIF. The image you select cannot exceed 1 Megabyte. You can't add a hyperlink to an image.

  Optionally, enter a description that appears when a user hovers over the image. The image must have a URL that Salesforce can access.
- The Salesforce HTML editor supports all languages that Salesforce Knowledge supports.
- The Salesforce HTML editor does not support JavaScript or Cascading Style Sheets (CSS).
- The Salesforce HTML editor is disabled for users who have accessibility mode enabled; it's replaced with a text box.

Supported HTML Tags and Attributes

The Salesforce HTML editor supports the tags listed in the following table.

<a>	<dt>	<q>
<abbr>		<samp>
<acronym>	<font>	<small>
<address>	<h1>	
<b>	<h2>	<strike>
<bdo>	<h3>	
<big>	<h4>	<sub>
<blockquote>	<h5>	<sup>
 	<h6>	<table>
<caption>	<hr>	<tbody>
<cite>	<i>	<td>
<code>	<img>	<tfoot>
<col>	<ins>	<th>
<colgroup>	<kbd>	<thead>
<dd>		<tr>

<del>		<tt>
<dfn>	<p>	
	<pre>	<var>
<dl>		

The tags can include the following attributes:

alt	face	size
background	height	src
border	href	style
class	name	target
colspan	rowspan	width

The attributes can include URLs that begin with the following:

- http:
- https:
- file:
- ftp:
- mailto:
- #
- / for relative links

SEE ALSO:

Rich Text Area Field Tips

Add Videos Using the HTML Editor

Add Videos Using the HTML Editor

Available in:Salesforce Classic.

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom fields:

- "Customize Application"

Before you can add videos to the HTML editor in a Knowledge article, make sure that your browser security settings allow you to embed multimedia content. Some browser security settings can block <iframe> elements. For example, if browser security is set to high in Internet Explorer 7, <iframe> content does not appear in the frame.

When the option to embed multimedia content is enabled during the setup of your application, users can cut and paste <iframe> HTML elements into the editor using one of two options:

1. Copy the `<iframe>` element from one of the approved video sites.
   Approved sites include Dailymotion, Vimeo, or YouTube.
2. Paste the code into the HTML editor by clicking one of these buttons:

Option	Description
	Lets you paste the `<iframe>` element into a text box on the Embed Multimedia Content dialog box. The frame and its contents are added to the editor window.
	Lets you paste the `<iframe>` element directly into the HTML code.

3. Click **Save**.

SEE ALSO:

Using the HTML Editor

Adding Code Samples to Posts

You can paste code samples into your posts in Chatter Answers and Ideas.

When you enable Chatter Answers and Ideas, users can add code samples to their posts. Code can be copied from any text editor and pasted into the body of a post with the formatting preserved.

1. Copy the code sample from a text editor to your clipboard.
2. In the editor, enter the text for your posts and click the  button to add the code sample.
3. Paste the code sample into the `Add a code sample` text box and click **OK**.
   The code sample appears in the body of the post with its formatting intact.

Change the Custom Field Type

Note: If you encrypt a custom field using Shield Platform Encryption, you can't change the field type.

1. From the management settings for the field's object, go to Fields.

   Note: For fields on Salesforce Knowledge article types, from Setup, enter *Knowledge Article Types* in the Quick Find box, select **Knowledge Article Types**, and then select an article type.

2. Click **Edit** next to the custom field you want to change.
3. Click **Change Field Type**.
4. Select a new data type and click **Next**.
5. Enter a field label, name, and any other attributes, and then save your changes.

SEE ALSO:

Notes on Changing Custom Field Types
Custom Field Types
Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Standard Objects are not available in **Database.com**

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To change custom fields:
- "Customize Application"

Notes on Changing Custom Field Types

Consider the following before converting fields:

- Only convert custom fields for which no data exists or you risk losing your data. Changing the data type of an existing custom field can cause data loss in the following situations:
  - Changing to or from type Date or Date/Time
  - Changing to Number from any other type
  - Changing to Percent from any other type
  - Changing to Currency from any other type
  - Changing from Checkbox to any other type
  - Changing from Picklist (Multi-Select) to any other type
  - Changing to Picklist (Multi-Select) from any other type

    Currently defined picklist values are retained when you change a picklist to a multi-select picklist. If records contain values that are not in the picklist definition, those values will be deleted from those records when the data type changes.

  - Changing from Auto Number to any other type
  - Changing to Auto Number from any type except Text
  - Changing from Text to Picklist
  - Changing from Text Area (Long) to any type except Email, Phone, Text, Text Area, or URL
- If data is lost, any list view based on the custom field will be deleted, and assignment and escalation rules may be affected.
- If you change the data type of any custom field that is used for lead conversion, that lead field mapping will be deleted.
- If you change the data type of a custom field that is set as an external ID, choosing a data type other than text, number, or email will cause the field to no longer act as an external ID.

- The option to change the data type of a custom field is not available for all data types. For example, existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type.
- In Salesforce Knowledge article types, the file field type can't be converted into other data types.
- You cannot change the data type of a custom field that is referenced by other items in Setup, like Visualforce pages, Apex code, Process Builder, or flows.
- Changing a custom field type may require changing a large number of records at once. To process these changes efficiently, your request may be queued and you may receive an email notification when the process has completed.
- Before changing a custom field's type, make sure that it isn't the target of a workflow field update or referenced in a field update formula that would be invalidated by the new type.

The following data types have additional restrictions when you convert them:

Data Type	Description
Auto Number	The data in any auto-number field remains unchanged if you convert it into a text field. Also, you can safely convert a text custom field into an auto-number field without losing your data. Converting an auto-number field into any other data type results in data loss. Auto-number fields can contain a maximum of 30 characters. Before converting a text custom field into an auto-number field, change any records that contain more than 30 characters in that field.
Formula	Formula fields are special read-only fields that cannot be converted to any other data type. Likewise, you cannot convert any other field type into a formula field.
Picklist	Changing your custom picklists into custom checkboxes is simple. If you select Checkbox as the new data type, you can choose which picklist values to map to checked boxes and unchecked boxes. You can change custom picklists into multi-select picklists without losing any data. Since your records only contain a single value of that picklist, that value will still be selected but users can select additional values.
Relationships	• You can convert relationship fields to nonrelationship fields and vice versa, but only on external objects. • If your organization has a large number of records, Salesforce displays a waiting page after you have requested to change a master-detail into a lookup relationship or a lookup into a master-detail relationship. • After you have created a roll-up summary field on an object, you cannot convert the object's master-detail relationship into a lookup relationship. • A lookup cannot be converted to a master detail relationship if there are any existing records on the object that have a null value set for the lookup relationship. • If you are converting a master-detail relationship to a lookup for a custom object on the "detail" side, the organization-wide default for the object is automatically updated to Public

Data Type	Description
	Read/Write. Similarly, converting a lookup to a master-detail-relationship changes the organization-wide default to Controlled by Parent
Text Area (Long)	When you convert a long text area field to an Email, Phone, Text, Text Area, or URL type field, the data in your records is truncated to the first 255 characters of the field.
Text Area (Rich)	You can only convert rich text area fields into long text area fields. Any images are deleted the next time the long text area field is saved. After converting, markup is hidden in the long text area field but it is not removed from the record until you save the record. That way, the markup can be restored if you change your mind.

SEE ALSO:

Change the Custom Field Type

Define Default Field Values

To define a default field value:

1. Begin by creating a custom field; see Create Custom Fields on page 125. You can also define a default value for an existing custom field; see Edit Custom Fields on page 143.
2. Choose the type of field to create and click **Next**. For a list of the types available for default values, see About Default Field Values on page 167.
3. Enter the attributes for the field.
4. Enter a default value or define a formula to calculate the default value.

   Note: You can define a formula for default values only where appropriate. For example, the default value options for picklist and checkbox fields are limited to the options available for those types of fields, such as `Checked`, `Unchecked`, or `Use first value as default value`.
5. Click **Next**.
6. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
7. Choose the page layouts that should display the field. The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is automatically added to the bottom of the user detail page.
8. Click **Save** to finish or **Save & New** to create more custom fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view default field values:
- "View Setup and Configuration"

To define or change default field values:
- "Customize Application"

Note: You must specify a default value for required campaign member custom fields.

Don't assign default values to fields that are both required and unique, because uniqueness errors can result.

SEE ALSO:

About Default Field Values

Default Field Value Considerations

Useful Default Field Value Formulas

About Default Field Values

Use default field values to make your users more productive by reducing the number of fields they need to fill in manually. Default field values automatically insert the value of a custom field when a new record is created. A default value can be based on a formula for some types of fields or exact values such as `Checked` or `Unchecked` for checkbox fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

After you have defined default values:

1. The user chooses to create a new record.
2. Default field value is executed.
3. Salesforce displays the edit page with the default field value pre-populated.
4. The user enters the fields for the new record.
5. The user saves the new record.

The user can change the field's value but the initial default field value is only executed once, during record creation. For example, you can set the default field value on a custom lead field to seven days after the creation date to signify when to contact the lead again. You can change this value later, but you cannot automatically restore the value that was seven days after the creation date.

Set up default field values for the following types of custom fields:

- Checkbox
- Currency
- Date
- Date/Time
- Email
- Number
- Percent
- Phone
- Picklist (use the default option when setting up the picklist)
- Text
- Text Area
- URL

For a description of these types, see Custom Field Types on page 129.

SEE ALSO:

Define Default Field Values
Default Field Value Considerations

Useful Default Field Value Formulas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view default field values:
- "View Setup and Configuration"

To define or change default field values:
- "Customize Application"

Maximum Discount Rate

Your organization my apply different discount rates to opportunities based on the department of the person creating the opportunity. Use the following example to set a default value for a custom field called `Discount Rate` on opportunities.

```
CASE(User.Department, "IT", 0.25, "Field", 0.15, 0)
```

In this example, the formula inserts a discount rate of 25% on any opportunity created by a user in the "IT" department or 15% on any opportunity created by someone in the "Field" department. A zero is applied if the creator does not belong to either of these departments. This is a custom percent field on opportunities that uses the standard user field `Department`.

Product Language

You may want to associate a product with its language so that your users know the type of documentation or adapter to include. Use the following default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is "Japanese" if the user's country is "Japan" and "English" if the user's country is "US." If neither is true, the default value "unknown" is inserted into the `Product Language` field.

```
CASE($User.Country , "Japan", "Japanese", "US", "English","unknown")
```

Tax Rate

Use this default value formula to set the tax rate of an asset based on the user's city. Create a custom percent field with the following default value:

```
IF($User.City = "Napa", 0.0750,
 IF($User.City = "Paso Robles", 0.0725,
  IF($User.City = "Sutter Creek", 0.0725,
   IF($User.City = "Los Olivos", 0.0750,
    IF($User.City = "Livermore", 0.0875, null
    )
   )
  )
 )
)
```

In this example, a tax rate of 8.75% is applied to an asset when the user's address is in the city of Livermore. When none of the cities listed apply, the `Tax Rate` field is empty. You can also use the `Tax Rate` field in formulas to automatically calculate taxable amounts and final sales prices.

SEE ALSO:

- Define Default Field Values
- Default Field Value Considerations

Default Field Value Considerations

Default field values automatically insert the value of a custom field when a new record is created. A default value can be based on a formula for some types of fields or exact values such as `Checked` or `Unchecked` for checkbox fields. Review the following considerations before incorporating default field values in your organization.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- If a default value is based on the value of a merge field, Salesforce uses the value of the merge field at the time the default value is executed. If the value of the merge field changes later, the default value is not updated.
- Users can change or remove the default field value on a record.
- Don't assign default values to fields that are both required and unique, because uniqueness errors can result.
- If you make an activity custom field universally required, you must also provide a default value.
- If an activity custom field is unique, you cannot provide a default value.
- Default field values are different from formula fields in the following ways: they are only executed once, at record creation; they are not read only; and the user can change the value but cannot restore the default field value.
- Since the default value is inserted before users enter any values in the new record, you cannot use the fields on the current record to create a default field value. For example, you cannot create a default field value on a contact that uses the first initial and last name because those values are not available when you click **New** to create a contact record. However, you can use the record type because it is selected before the record edit page displays.
- To apply a different default value for different record types, use the record type as a merge field in a CASE function within the default field value setup.
- Fields that are not visible to the user due to field-level security are still available in the formula for a default field value.

- Connect Offline and Salesforce for Outlook do not display default values. However, Salesforce inserts the default values when a user syncs unless the user entered a value.
- Default field values are not available in the Self-Service portal.
- Lead conversion, Web-to-Lead, and Web-to-Case do not execute default field values.

Note: You can define a formula for default values only where appropriate. For example, the default value options for picklist and checkbox fields are limited to the options available for those types of fields, such as `Checked`, `Unchecked`, or `Use first value as default value`.

SEE ALSO:

Define Default Field Values

Useful Default Field Value Formulas

Validation Rules

Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

After you have defined validation rules:

1. The user chooses to create a new record or edit an existing record.
2. The user clicks **Save**.
3. All validation rules are verified.
   - If all data is valid, the record is saved.
   - If any data is invalid, the associated error message displays without saving the record.
4. The user makes the necessary changes and clicks **Save** again.

You can specify the error message to display when a record fails validation and where to display it. For example, your error message can be "The close date must occur after today's date." You can choose to display it near a field or at the top of the page. Like all other error messages, validation rule errors display in red text and are preceded by the word "Error".

Important: Validation rules apply to new and updated records for an object, even if the fields referenced in the validation rule are not included in a page layout or an API call. Validation rules don't apply if you create new records for an object with Quick Create. If your organization has multiple page layouts for the object on which you create a validation rule, verify that the validation rule functions as intended on each layout. If your organization has any integrations that use this object, verify that the validation rule functions as intended for each integration.

SEE ALSO:

Define Validation Rules

Validation Rules Fields

Examples of Validation Rules

Managing Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

From the validation rules page you can:

- Define a validation rule.
- Click **Edit** next to a rule name to update the rule fields.
- Delete a validation rule.
- Click a validation rule name to view more details or to clone the rule.
- Activate a validation rule.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

SEE ALSO:

Validation Rules

Examples of Validation Rules

Define Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

Before creating validation rules, review the Validation Rule Considerations.

1. From the management settings for the relevant object, go to Validation Rules.
2. In the Validation Rules related list, click **New**.
3. Enter the properties of your validation rule.
4. To check your formula for errors, click **Check Syntax**.
5. Click **Save** to finish or **Save & New** to create additional validation rules.

Note: The detail page of a custom activity field does not list associated validation rules. To edit the validation rule for a custom activity field, select the validation rule from Setup by entering *Activities* in the Quick Find box, then selecting **Activities** and choose **Task Validation Rules** or **Event Validation Rules**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view field validation rules:
- "View Setup and Configuration"

To define or change field validation rules:
- "Customize Application"

SEE ALSO:

Validation Rules

Clone Validation Rules

Tips for Writing Validation Rules

Examples of Validation Rules

Clone Validation Rules

1. From the management settings for the relevant object, go to Validation Rules.
2. In the Validation Rules related list, click the name of the validation rule.
3. Click **Clone**.
4. Define the new rule based on the original rule.
5. Click **Save** to finish or **Save & New** to create additional validation rules.

Note: The detail page of a custom activity field does not list associated validation rules. To edit the validation rule for a custom activity field, select the validation rule from Setup by entering *Activities* in the Quick Find box, then selecting **Activities** and choose **Task Validation Rules** or **Event Validation Rules**.

SEE ALSO:

Define Validation Rules
Validation Rules Fields
Activate Validation Rules

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view field validation rules:
- "View Setup and Configuration"

To define or change field validation rules:
- "Customize Application"

Activate Validation Rules

1. From the management settings for the relevant object, go to Validation Rules.
2. Click **Edit** next to the rule you want to activate.
3. To activate the rule, select **Active**, and save your changes.
4. To deactivate the rule, deselect **Active**, and save your changes.

Note: The detail page of a custom activity field does not list associated validation rules.

SEE ALSO:

Define Validation Rules
Validation Rules Fields
Find Object Management Settings

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view field validation rules:
- "View Setup and Configuration"

To define or change field validation rules:
- "Customize Application"

Validation Rules Fields

Field	Description
`Rule Name`	Unique identifier of up to 40 characters with no spaces or special characters such as extended characters.
`Active`	Checkbox that indicates if the rule is enabled.
`Description`	A 255-character or less description that distinguishes the validation rule from others. For internal purposes only.
`Error Condition Formula`	The expression used to validate the field. See Build a Formula Field and Formula Operators and Functions.
`Error Message`	The message that displays to the user when a field fails the validation rule. If your organization uses the Translation Workbench, you can translate the error message into the languages Salesforce supports. See Enable and Disable the Translation Workbench.
`Error Location`	Determines where on the page to display the error. To display the error next to a field, choose `Field` and select the field. If the error location is a field, the validation rule is also listed on the detail page of that field. If the error location is set to a field that is later deleted, to a field that is read only, or to a field that isn't visible on the page layout, Salesforce automatically changes the location to `Top of Page`. Note: Error messages can only be displayed at the top of the page in validation rules for case milestones and Ideas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

SEE ALSO:

Define Validation Rules

Tips for Writing Validation Rules

- Consider all the settings in your organization that can make a record fail validation, including assignment rules, field updates, field-level security, or fields hidden on a page layout.
- Be careful not to create contradicting validation rules for the same field; otherwise, users won't be able to save the record.

  Tip: A poorly designed validation rule can prevent users from saving valid data. Make sure you thoroughly test a validation rule before activating it. You can also use the debug log to monitor the details of your rule implementation.
- When referencing related fields in a validation formula, make sure those objects are deployed.
- Use the `RecordType.Id` merge field in your formula to apply different validations for different record types.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

- You don't have to begin a validation rule formula with the IF function. Any Boolean error condition expression works. For example:
  - Correct: `CloseDate < TODAY()`
  - Incorrect: `IF(CloseDate < TODAY(), TRUE, FALSE)`
- Keep in mind that when a validation rule contains the BEGINS or CONTAINS function, it processes blank fields as valid. For example, if you have a validation rule that tests whether the serial number of an asset begins with "3", all assets with a blank serial number are considered valid.
- When using a validation rule to ensure that a number field contains a specific value, use the ISBLANK function to include fields that don't contain any value. For example, to validate that a custom field contains a value of '1', use the following validation rule to display an error if the field is blank or any other number:

```
OR (ISBLANK (field__c), field__c<>1)
```

- Avoid using the `IsClosed` or `IsWon` opportunity merge fields in validation formulas. Instead, use the ISPICKVAL function to determine if the `Stage` contains the appropriate value. For example, the following validation formula makes a custom `Project Start Date` field required whenever the `Stage` is "Closed Won":

```
AND(ISPICKVAL(StageName, "Closed Won"),
   ISBLANK(Project_Start_Date__c))
```

- Simplify your validation formulas by using checkbox fields, which don't require any operator because they return true or false. For example, the following validation formula checks to be sure an opportunity has opportunity products using the `HasOpportunityLineItem` merge field before users can save a change to it:

```
NOT(OR(ISNEW(),HasOpportunityLineItem))
```

Tips for Writing Validation Rule Error Messages

- Give instructions. An error message like "invalid entry" doesn't tell the user what type of entry is valid. Write something more specific, such as "`Close Date` must be after today."
- Always include the field label. Users might not know what field is failing validation, especially if the error message appears at the top of the page.

  Note: When defining validation rules, you can set the error location to `Top of Page` or `Field`. If the error location is set to a field that is later deleted, to a field that is read only, or to a field that isn't visible on the page layout, Salesforce automatically changes the location to `Top of Page`.
- If you have a multilingual organization, translate your error messages. You can translate error messages using the Translation Workbench.
- Assign corresponding numbers to validation rules and their error messages. This allows you to identify the source of the error.

SEE ALSO:

Define Validation Rules

Validation Rule Considerations

Validation Rule Considerations

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value. Review these considerations before implementing validation rules in your organization.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

How Salesforce Processes Validation Rules

Salesforce processes rules in the following order:

1. Validation rules
2. Assignment rules
3. Auto-response rules
4. Workflow rules (with immediate actions)
5. Escalation rules

In addition:

- When one validation rule fails, Salesforce continues to check any additional validation rules on that field or any other field on the page and displays all appropriate error messages at once.
- If validation rules exist for activities and you create an activity during lead conversion, the lead converts but a task isn't created.
- Validation rules are only enforced during lead conversion if validation and triggers for lead conversion are enabled in your organization.
- Campaign hierarchies ignore validation rules.
- Salesforce runs validation rules before creating records submitted via Web-to-Lead and Web-to-Case, and only creates records that have valid values.
- Validation rules continue to run on individual records if the owner is changed. If the Mass Transfer tool is used to change the ownership of multiple records, however, validation rules won't run on those records.

Validation Rule Field Restrictions

Validation rule formulas don't or can't refer to:

- Compound fields, including addresses, first and last names, and dependent picklists and lookups
- Campaign statistic fields, including statistics for individual campaigns and campaign hierarchies
- Merge fields for auto-number or compound address fields such as `Mailing Address`

  Note: Merge fields for individual address fields, such as `Billing City`, are OK to use in validation rule formulas.

In addition, validation rules behave like this with regard to other fields and functions in Salesforce:

- The detail page of a custom activity field doesn't list associated validation rules.
- Because updates to records based on workflow rules and also on process scheduled actions don't trigger validation rules, workflow rules and some processes can invalidate previously valid fields.
- Process record updates on immediate actions fire validation rules.
- You can't create validation rules for relationship group members.
- Because roll-up summary fields aren't displayed on edit pages, you can use them in validation rules, but not as the error location.

Lookup Filters vs. Validation Rules

Validation rules and lookup filters achieve similar ends, but offer different advantages. Use a lookup filter if:

- You want to improve user efficiency by limiting the number of available options in a lookup search dialog.
- You want to improve user efficiency by automating filters on lookup search dialogs that your users manually set.

Use a validation rule if:

- You're close to the maximum number of lookup filters allowed.
- You must implement a complex business rule that requires you to use a formula. Formulas can reference fields that basic filter criteria can't reference, such as fields on the parent of the source object. Formulas can also use functions. For example, use `ISNEW` if the rule should only apply on record creation, or `ISCHANGED` if the rule should apply when a field changes.

SEE ALSO:

Define Validation Rules

Activate Validation Rules

Examples of Validation Rules

Examples of Validation Rules

Review examples of validation rules for various types of apps that you can use and modify for your own purposes. *Validation rules* verify that the data a user enters in a record meets the standards you specify before the user can save the record.

Use the following samples for validation rules in Salesforce and Force.com AppExchange apps, including:

IN THIS SECTION:

Sample Account Address Validation Rules

Sample Account Validation Rules

Sample Call Center Validation Rules

Sample Community Validation Rules

Sample Contact Validation Rules

Sample Cross Object Validation Rules

Sample Date Validation Rules

Sample Number Validation Rules

Sample Opportunity Management Validation Rules

Sample Quote Validation Rules

Sample User, Role, and Profile Validation Rules

Miscellaneous Sample Validation Rules

SEE ALSO:

Validation Rules

Define Validation Rules

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view field validation rules:

- "View Setup and Configuration"

To define or change field validation rules:

- "Customize Application"

Sample Account Address Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Canadian Billing Postal Code

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is in the correct format if `Billing Country` is Canada.

Formula:	`AND(`
	`OR(BillingCountry = "CAN", BillingCountry = "CA", BillingCountry = "Canada"),`
	`NOT(REGEX(BillingPostalCode, "((?i)[ABCEGHJKLMNPRSTVXY]\\d[A-Z]?\\s?\\d[A-Z]\\d)?"))`
	`)`

Error Message:	Canadian postal code must be in A9A 9A9 format.
Error Location:	**Billing Zip/Postal Code**

Billing Zip Code Is in Billing State

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid zip code in the US. If the zip code is not found in the Zip_Code__c object, or the `Billing State` does not match the corresponding State_Code__c in the Zip_Code__c object, an error is displayed.

Formula:	`VLOOKUP(`
	`$ObjectType.Zip_Code__c.Fields.City__c ,`
	`$ObjectType.Zip_Code__c.Fields.Name ,`
	`LEFT(BillingPostalCode,5)) <> BillingCity`

Error Message:	Billing Zip Code does not exist in specified Billing State.
Error Location:	**Billing Zip/Postal Code**

US Billing Zip Code

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is in 99999 or 99999-9999 format if `Billing Country` is USA or US.
Formula:	``` AND(OR(BillingCountry = "USA", BillingCountry = "US"), NOT(REGEX(BillingPostalCode, "\\d{5}(-\\d{4})?"))) ``` Note: This example uses the REGEX function; see Shipping Zip Code if you are not familiar with regular expressions.
Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	**Billing Zip/Postal Code**

Shipping Zip Code

Field	Value
Description:	Validates that the account `Shipping Zip/Postal Code` is in 99999 or 99999-9999 format if `Shipping Country` is USA or blank.
Formula:	

```
AND(
 OR(ShippingCountry = "USA", ISBLANK(ShippingCountry)),
  OR(
   AND(LEN(ShippingPostalCode) <>5,
      LEN(ShippingPostalCode) <> 10),
   NOT(CONTAINS("0123456789",
      LEFT( ShippingPostalCode, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 2, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 3, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 4, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 5, 1))),
   AND(
    LEN(ShippingPostalCode) = 10,
   OR(
    MID( ShippingPostalCode , 6, 1) <> "-",
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 7, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 8, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 9, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 10, 1)))
   )
  )
 )
)
```

Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Tip: You can also validate zip codes using a regular expression; for an example of a formula using a regular expression, see REGEX on page 275.

Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	**Shipping Zip/Postal Code**

Valid Billing State (US)

Field	Value
Description:	Validates that the account `Billing State/Province` is a valid two-character abbreviation if `Billing Country` is US, USA, or blank.

Formula:	`AND (`
	`OR(BillingCountry = "US", BillingCountry="USA",`
	`ISBLANK(BillingCountry)),`
	`OR(`
	`LEN(BillingState) < 2,`
	`NOT(`
	`CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:" &`
	`"IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:" &`
	`"NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:" &`
	`"WA:WV:WI:WY", BillingState)`
	`)))`
	
	Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Error Message:	A valid two-letter state code is required.
Error Location:	`Billing State/Province`

Valid Billing Province (Canada)

Field	Value
Description:	Validates that the account `Billing State/Province` is a valid two-character abbreviation if `Billing Country` is CA or CAN.

Formula:	`AND (`
	`OR(BillingCountry = "CA", BillingCountry="CAN"),`
	`OR(`
	`LEN(BillingState) < 2,`
	`NOT(`
	`CONTAINS("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT", BillingState)`
	`)))`

Error Message:	A valid two-letter province code is required.
Error Location:	`Billing State/Province`

Valid Shipping State

Field	Value
Description:	Validates that the account `Shipping State/Province` is a valid two-character abbreviation if `Shipping Country` is US, USA, or blank.

Formula:	`AND (`
	`OR(ShippingCountry = "US", ShippingCountry="USA",`
	`ISBLANK(ShippingCountry)),`
	`OR(`
	`LEN(ShippingState) < 2,`
	`NOT(`
	`CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:" &`
	`"IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:" &`
	`"NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:" &`
	`"WA:WV:WI:WY", ShippingState)`
	`)))`
	
	Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Error Message:	A valid two-letter state abbreviation is required.
Error Location:	`Shipping State/Province`

Valid Shipping Province (Canada)

Field	Value
Description:	Validates that the account `Shipping State/Province` is a valid two-character abbreviation, if `Billing Country` is CA or CAN.

Formula:	`AND (`
	`OR(ShippingCountry = "CA", ShippingCountry="CAN"),`
	`OR(`
	`LEN(ShippingState) < 2,`
	`NOT(`
	`CONTAINS("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT", ShippingState)`
	`))`

Error Message:	A valid two-letter province abbreviation is required.
Error Location:	`Shipping State/Province`

Valid Billing Country

Field	Value
Description:	Validates that the account `Billing Country` is a valid ISO 3166 two-letter code.

Formula:	`OR(`
	`LEN(BillingCountry) = 1,`
	`NOT(`
	`CONTAINS(`
	`"AF:AX:AL:DZ:AS:AD:AO:AI:AQ:AG:AR:AM:" &`
	`"AW:AU:AZ:BS:BH:BD:BB:BY:BE:BZ:BJ:BM:BT:BO:" &`
	`"BA:BW:BV:BR:IO:BN:BG:BF:BI:KH:CM:CA:CV:KY:" &`
	`"CF:TD:CL:CN:CX:CC:CO:KM:CG:CD:CK:CR:CI:HR:" &`
	`"CU:CY:CZ:DK:DJ:DM:DO:EC:EG:SV:GQ:ER:EE:ET:FK:" &`
	`"FO:FJ:FI:FR:GF:PF:TF:GA:GM:GE:DE:GH:GI:GR:GL:" &`
	`"GD:GP:GU:GT:GG:GN:GW:GY:HT:HM:VA:HN:HK:HU:" &`
	`"IS:IN:ID:IR:IQ:IE:IM:IL:IT:JM:JP:JE:JO:KZ:KE:KI:" &`
	`"KP:KR:KW:KG:LA:LV:LB:LS:LR:LY:LI:LT:LU:MO:MK:" &`
	`"MG:MW:MY:MV:ML:MT:MH:MQ:MR:MU:YT:MX:FM:MD:MC:" &`
	`"MC:MN:ME:MS:MA:MZ:MM:MA:NR:NP:NL:AN:NC:NZ:NI:" &`
	`"NE:NG:NU:NF:MP:NO:OM:PK:PW:PS:PA:PG:PY:PE:PH:" &`
	`"PN:PL:PT:PR:QA:RE:RO:RU:RW:SH:KN:LC:PM:VC:WS:" &`
	`"SM:ST:SA:SN:RS:SC:SL:SG:SK:SI:SB:SO:ZA:GS:ES:" &`
	`"LK:SD:SR:SJ:SZ:SE:CH:SY:TW:TJ:TZ:TH:TL:TG:TK:" &`
	`"TO:TT:TN:TR:TM:TC:TV:UG:UA:AE:GB:US:UM:UY:UZ:" &`
	`"VU:VE:VN:VG:VI:WF:EH:YE:ZM:ZW",`
	`BillingCountry)))`

Error Message:	A valid two-letter country code is required.
Error Location:	`Billing Country`

Sample Account Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Account Number Is Numeric

Field	Value
Description:	Validates that the `Account Number` is numeric if not blank.

Formula:	`AND(`
	`  ISBLANK(AccountNumber),`
	`  NOT(ISNUMBER(AccountNumber))`
	`)`

Error Message:	Account Number is not numeric.
Error Location:	`Account Number`

Account Number Length

Field	Value
Description:	Validates that the `Account Number` is exactly seven digits (if it is not blank). The number seven is simply illustrative. You can change this to any number you like.

Formula:	`AND(`
	`  ISBLANK(AccountNumber),`
	`  LEN(AccountNumber) <> 7`
	`)`

Error Message:	Account Number must be seven digits.
Error Location:	`Account Number`

Annual Revenue Range

Field	Value
Description:	Validates that the account `Annual Revenue` is not negative and does not exceed $100 billion. This limit is designed to catch typos.

Formula:	`OR(`
	`  AnnualRevenue < 0,`
	`  AnnualRevenue > 100000000000`
	`)`

Error Message:	Annual Revenue cannot exceed 100 billion.
Error Location:	`Annual Revenue`

Sample Call Center Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Conditionally Require Description When Case Reason is "Other"

Field	Value
Description:	Validates that a custom field called `Other Reason` contains a value if a case has a `Case Reason` of "Other."

Formula:	`AND(`
	`  ISPICKVAL( Reason, "Other" ),`
	`  ISBLANK(Other_Reason__c)`
	`)`

Error Message:	Description of Other Reason is required.
Error Location:	`Other Reason`

Prevent Open Cases from Being Reset to New

Field	Value
Description:	If a case is already open, prevents the `Status` from being changed back to "New."

Formula:	`AND(`
	`  ISCHANGED( Status ),`
	`  NOT(ISPICKVAL(PRIORVALUE( Status ), "New")),`
	`  ISPICKVAL( Status, "New")`
	`)`

Error Message:	Open case Status cannot be reset to New.
Error Location:	`Status`

Restrict Status of Re-Opened Cases

Field	Value
Description:	Validates that the case `Status` is "Re-opened" when a closed case is opened again.

Formula:	`AND(`
	`  ISCHANGED( Status ),`
	`  OR(`
	`  ISPICKVAL(PRIORVALUE( Status ), "Closed"),`
	`  ISPICKVAL(PRIORVALUE( Status ),`
	`  "Closed in SSP")),`
	`  NOT( ISPICKVAL( Status, "Re-Opened"))`
	`)`

Error Message:	Closed case can only be changed to "Re-opened."
Error Location:	`Status`

Prevent Case Milestone Completion After Cases Are Closed

Field	Value
Description:	Validates that a milestone's `Completion Date` can't occur after the case's `Status` is Closed.
Formula:	`Case.IsClosed = true`
Error Message:	You can't complete a milestone after a case is closed.
Error Location:	`Top of Page`

Prevent Case Milestone Completion Before Case Creation Dates

Field	Value
Description:	Validates that the milestone's `Completion Date` has occurred after the case's `Date/Time Opened`.
Formula:	`CompletionDate >= Case.CreatedDate && CompletionDate <= Case.ClosedDate`
Error Message:	The milestone Completion Date must occur after the date the case was created and before the case was closed.
Error Location:	`Top of Page`

Sample Community Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Preventing Offensive Language in Questions

Field	Value
Description:	Prevents users from entering offensive language in the `Title` and `Description` fields when asking a question.
Formula:	`OR(CONTAINS(Title, 'darn'), CONTAINS(Body, 'darn'))`
Error Message:	Question title or description contains offensive language.

Preventing Offensive Language in Replies

Field	Value
Description:	Prevents users from entering offensive language when replying to a question.
Formula:	`OR(CONTAINS(Body, 'darn'), CONTAINS(Body, 'dang'))`
Error Message:	Reply contains offensive language.

Preventing Offensive Language in Ideas

Field	Value
Description:	Prevents users from entering offensive language in the `Title` and `Description` fields when posting an idea.
Formula:	`OR(CONTAINS(Title, 'darn'), CONTAINS(Body, 'darn'))`
Error Message:	Idea title or description contains offensive language.

Preventing Offensive Language in Idea Comments

Field	Value
Description:	Prevents users from entering offensive language when posting a comment.
Formula:	`OR(CONTAINS(CommentBody , 'darn'), CONTAINS(CommentBody, 'dang'))`
Error Message:	Comment contains offensive language.

Sample Contact Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Mailing Address Fields Are Required

Field	Value
Description:	Validates that the contact `Mailing Street`, `Mailing City`, and `Mailing Country` are provided.
Formula:	`OR( ISBLANK( MailingStreet ), ISBLANK( MailingCity ), ISBLANK( MailingCountry ) )`
Error Message:	Mailing Street, City, and Country are required.
Error Location:	`Top of Page`

Mailing Street Is Required

Field	Value
Description:	Validates that the contact `Mailing Street` is provided.
Formula:	`ISBLANK( MailingStreet )`
Error Message:	Mailing Street is required.
Error Location:	`Mailing Street`

Mailing Zip Code

Field	Value
Description:	Validates that the contact `Mailing Zip/Postal Code` is in 99999 or 99999-9999 format if `Mailing Country` is USA or blank.
Formula:	

```
AND(
 OR(MailingCountry = "USA", ISBLANK(MailingCountry)),
  OR(
   AND(LEN(MailingPostalCode) <>5,
   LEN(MailingPostalCode) <> 10),
   NOT(CONTAINS("0123456789",
      LEFT( MailingPostalCode, 1))),
   NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 2, 1))),
   NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 3, 1))),
   NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 4, 1))),
   NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 5, 1))),
   AND(
    LEN(MailingPostalCode) = 10,
   OR(
    MID( MailingPostalCode , 6, 1) <> "-",
    NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 7, 1))),
    NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 8, 1))),
    NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 9, 1))),
    NOT(CONTAINS("0123456789",
      MID( MailingPostalCode , 10, 1)))
   )
  )
 )
)
```

Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Tip: You can also validate zip codes using a regular expression; for an example of a formula using a regular expression, see REGEX on page 275.

Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	`Mailing Zip/Postal Code`

Phone Number Has International Format

Field	Value
Description:	Validates that the `Phone` number begins with a plus sign (+) for country code. Note that this validation rule conflicts with the ten-digit rule.
Formula:	`LEFT(Phone, 1) <> "+"`
Error Message:	Phone number must begin with + (country code).
Error Location:	`Phone`

US Phone Number Has Ten Digits

Field	Value
Description:	Validates that the `Phone` number is in (999) 999-9999 format. This works by using the REGEX function to check that the number has ten digits in the (999) 999-9999 format.
Formula:	`NOT(REGEX(Phone, "\\D*?(\\d\\D*?){10}"))`
Error Message:	US phone numbers should be in this format: (999) 999-9999.
Error Location:	`Phone`

Sample Cross Object Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Discounts Must Be Within Range

This example consists of three validation rules on opportunity products. The examples below work together to help you manage discount amounts for products and require a custom percent field on opportunity products called `Line Discount`. The examples below also require you to use price books and customize the `Product Family` field to include the following values: *`Software`*, *`Consulting`*, and *`Training`*.

Software Discounts

Field	Value
Description:	Prevents users from saving software products with a discount over 10 percent.
Formula:	`AND(Line_Discount__c > 0.10, ISPICKVAL(Product2.Family, "Software"))`

Field	Value
Error Message:	The discount must be 10% or less for software products.
Error Location:	`Line Discount`

Consulting Discounts

Field	Value
Description:	Prevents users from saving consulting products with a discount over 15 percent.
Formula:	`AND(Line_Discount__c > 0.15, ISPICKVAL(Product2.Family, "Consulting"))`
Error Message:	The discount must be 15% or less for consulting products.
Error Location:	`Line Discount`

Training Discounts

Field	Value
Description:	Prevents users from saving training products with a discount over 20 percent.
Formula:	`AND(Line_Discount__c > 0.20, ISPICKVAL(Product2.Family, "Training"))`
Error Message:	The discount must be 20% or less for training products.
Error Location:	`Line Discount`

Prevent Changing Opportunity Products on Closed Opportunities

This example consists of two validation rules: one on opportunity products and another on opportunities.

Field	Value
Description:	Prevents users from editing opportunity products after an opportunity is closed. Create the following validation rule example on opportunity products.
Formula:	`OR(ISPICKVAL(Opportunity.StageName, "Closed Won"), ISPICKVAL(Opportunity.StageName, "Closed Lost"))`
Error Message:	Cannot change opportunity products for closed opportunities.

Field	Value
Error Location:	`Top of Page`

The following validation rule is on opportunities.

Field	Value
Description:	Prevents users from deleting opportunity products after an opportunity is closed. Create the following validation rule example on opportunities. It uses a custom roll-up summary field on opportunities that counts the number of opportunity products on an opportunity.
Formula:	`AND(OR(ISPICKVAL(StageName, "Closed Won"), ISPICKVAL(StageName, "Closed Lost")), Number_of_Line_Items__c < PRIORVALUE(Number_of_Line_Items__c) )`
Error Message:	Cannot delete opportunity products for closed opportunities.
Error Location:	`Top of Page`

Prevent Saving a Case When Account Does Not Have Support

Field	Value
Description:	Prevents users from saving a case for an account that does not have support. This example assumes you have a custom checkbox field on accounts called `Allowed Support` that tracks if the account has support.
Formula:	`Account.Allowed_Support__c = FALSE`
Error Message:	Unable to create cases for this account because it is not signed up for support.
Error Location:	`Top of Page`

Prevent Saving a Case When Contact is No Longer with the Company

Field	Value
Description:	Prevents users from saving an open case associated with a contact that is no longer with the company. This example uses a custom checkbox field on contacts called `No Longer With Company`.

Field	Value
Formula:	`AND(Contact.Not_Longer_With_Company__c, NOT(IsClosed))`
Error Message:	Unable to save this case because the related contact is no longer with the company. To continue, choose another contact.
Error Location:	`Contact Name`

Sample Date Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Date Must Be a Weekday

Field	Value
Description:	Validates that the value of a custom date field is a weekday (not Saturday or Sunday).
Formula:	`CASE(MOD( My_Date__c - DATE(1900, 1, 7), 7), 0, 0, 6, 0, 1) = 0`
Error Message:	Date must be a weekday.
Error Location:	`My Date`

Date Must Be a Weekend Day

Field	Value
Description:	Validates that the value of a custom date field is a Saturday or Sunday.
Formula:	`CASE( MOD( My_Date__c - DATE(1900, 1, 7), 7), 0, 1, 6, 1, 0) = 0`
Error Message:	Date must be a weekend day.
Error Location:	`My Date`

Date Must Be in the Current Month

Field	Value
Description:	Validates that a custom date field contains a date within the current month and year.

Formula:	`OR (`
	`YEAR( My_Date__c ) <> YEAR ( TODAY() ),`
	`MONTH( My_Date__c ) <> MONTH ( TODAY() )`
	`)`

Error Message:	Date must be in the current month.
Error Location:	`My Date`

Date Must Be in the Current Year

Field	Value
Description:	Validates that a custom date field contains a date within the current year.
Formula:	`YEAR( My_Date__c ) <> YEAR ( TODAY() )`
Error Message:	Date must be in the current year.
Error Location:	`My Date`

Date Must Be the Last Day of the Month

Field	Value
Description:	Validates whether a custom field called `My Date` is the last day of the month. To do this, it determines the date of the first day of the next month and then subtracts 1 day. It includes special case logic for December.

Formula:	`DAY(My_Date__c) <>`
	`IF(Month(My_Date__c)=12, 31,`
	`DAY(DATE(YEAR(My_Date__c),MONTH(My_Date__c)+1,1) - 1))`

Error Message:	Date must be the last day of the month.
Error Location:	`My Date`

Date Must Be Within One Year of Today

Field	Value
Description:	Validates whether a custom field called `Follow-Up Date` is within one year of today's date. This example assumes a 365 day year. (It does not handle leap years.)
Formula:	`Followup_Date__c - TODAY() > 365`
Error Message:	Follow-Up Date must be within one year of today.
Error Location:	`Follow-Up Date`

Day of Month Cannot Be Greater Than 15

Field	Value
Description:	Validates that a custom field called `Begin Date` contains a date in the first 15 days of the specified month.
Formula:	`DAY( Begin_Date__c ) > 15`
Error Message:	Begin Date cannot be after the 15th day of month.
Error Location:	`Begin Date`

End Date Cannot Be Before Begin Date

Field	Value
Description:	Validates that a custom field called `End Date` does not come before another custom field called `Begin Date`.
Formula:	`Begin_Date__c > End_Date__c`
Error Message:	End Date cannot be before Begin Date.
Error Location:	`Begin Date`

Expiration Date Cannot Be Before Close Date

Field	Value
Description:	Validates that a custom field called `Expiration Date` does not come before `Close Date`.
Formula:	`Expiration_Date__c < CloseDate`
Error Message:	Expiration Date cannot be before Close Date.
Error Location:	`Expiration Date`

Sample Number Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Time Cards Must Total 40 Hours

Field	Value
Description:	Ensures that users cannot save a time card record with more than 40 hours in a work week. This example requires five custom fields on your custom object, one for each day of work.
Formula:	`Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40`
Error Message:	Your total hours cannot exceed 40.
Error Location:	`Top of Page`

Number Cannot Be Negative

Field	Value
Description:	Validates that a custom field called `Hours Worked` is not a negative number.
Formula:	`Hours_Worked__c < 0`
Error Message:	Hours Worked cannot be less than zero.
Error Location:	`Hours Worked`

Number Must Be Even

Field	Value
Description:	Validates that a custom field called `Ark Passengers` is a non-negative even number.
Formula:	`OR( Ark_Passengers__c < 0, MOD( Ark_Passengers__c, 2) <> 0 )`
Error Message:	Ark Passengers must be a positive even number.
Error Location:	`Ark Passengers`

Number Must Be Odd

Field	Value
Description:	Validates that a custom field called `Socks Found` is a non-negative odd number.
Formula:	`OR( Socks_Found__c < 0, MOD( Socks_Found__c, 2) = 0 )`
Error Message:	Socks Found must be an odd number.
Error Location:	`Socks Found`

Number Must Be a Multiple of Five

Field	Value
Description:	Validates that a custom field called `Multiple of 5` is a multiple of five.
Formula:	`MOD( Multiple_of_5__c, 5) <> 0`
Error Message:	Number must be a multiple of five.
Error Location:	`Multiple of 5`

Number Must Be an Integer

Field	Value
Description:	Validates that a custom field called `My Integer` is an integer.
Formula:	`FLOOR( My_Integer__c) <> My_Integer__c`
Error Message:	This field must be an integer.
Error Location:	`My Integer`

Number Must Be Between -50 and 50

Field	Value
Description:	Validates that a custom field called `Volume` is between -50 and 50.
Formula:	`ABS( Volume__c) > 50`
Error Message:	Volume must be between -50 and 50.
Error Location:	`Volume`

Number Range Validation

Field	Value
Description:	Validates that the range between two custom fields, `Salary Min` and `Salary Max`, is no greater than $20,000.
Formula:	`(Salary_Max__c - Salary_Min__c) > 20000`
Error Message:	Salary range must be within $20,000. Adjust the Salary Max or Salary Min values.
Error Location:	`Salary Max`

Percentage Must Be Between Zero and 100

Field	Value
Description:	Validates that a custom field called `Mix Pct` is between 0 and 100%. Note that percent fields are expressed divided by 100 in formulas (100% is expressed as 1; 50% is expressed as 0.5).

Formula:	`OR(`
	`  Mix_Pct__c > 1.0,`
	`  Mix_Pct__c < 0.0`
	`)`

Error Message:	Mix Pct must be between 0 and 100%.
Error Location:	`Mix Pct`

Sample Opportunity Management Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Conditionally-Required Field Based on Opportunity Stage

Field	Value
Description:	Validates that a custom field called `Delivery Date` is provided if an opportunity has advanced to the Closed Won or Negotiation/Review stage.

Formula:	`AND (`
	`  OR (`
	`  ISPICKVAL(StageName, "Closed Won"),`
	`  ISPICKVAL(StageName,`
	`"Negotiation/Review")),`
	`ISBLANK(Delivery_Date__c)`
	`)`

Error Message:	Delivery Date is required for this stage.
Error Location:	`Delivery Date`

Close Date Cannot Be Prior to Current Month

Field	Value
Description:	Validates that the `Close Date` of an opportunity is not within a month prior to the current month. Note the use of ISNEW and ISCHANGED in this formula to ensure the condition is only checked when the opportunity is being created or the `Close Date` field is modified subsequently.

Formula:	`AND(`
	`  OR (`
	`  ISNEW(),`
	`  ISCHANGED( CloseDate )),`
	`  CloseDate < DATE( YEAR(TODAY()), MONTH(TODAY()), 1) )`

Error Message:	Close Date cannot be prior to current month.
Error Location:	`Close Date`

Close Date Must Be a Future Date

Field	Value
Description:	Ensures that users do not change the `Close Date` of an opportunity to a day in the past.
Formula:	`SampleDate < TODAY()`
Error Message:	Close Date cannot be a day in the past.
Error Location:	`Close Date`

Discounts on Opportunities

Field	Value
Description:	Validates that a custom discount percent field is between 0 and 40%.
Formula:	`OR(Discount_Rate__c < 0, Discount_Rate__c > 0.40)`
Error Message:	The Discount Rate must not exceed 40%.
Error Location:	`Discount Rate`

High-Value Opportunity Must Be Approved Before Closed

Field	Value
Description:	Opportunities with amounts greater than $50,000 require that a custom checkbox field called `Approved` is checked in order to change the stage to Closed Won or Closed Lost. To automate this, set field-level security on the `Approved` checkbox so that it can only be checked via a custom approval process (Enterprise Edition, Unlimited Edition, or Performance Edition).

Formula:	`AND(`
	`  OR(`
	`  ISPICKVAL(StageName,"Closed Won"),`
	`  ISPICKVAL(StageName,"Closed Lost")),`
	`(Amount > 50000),`
	`NOT(ISPICKVAL(Approval_Status__c ,"Approved")))`

Error Message:	All high-value opportunities must be approved for closure. Click the Request Close button.
Error Location:	`Top of Page`

Opportunity Amount Cannot Exceed $10 Million

Field	Value
Description:	Validates that opportunity `Amount` is positive and no more than $10 million. This limit is designed to catch typos.

Formula:	`OR(`
	`  Amount < 0,`
	`  Amount > 10000000`
	`)`

Error Message:	Amount cannot exceed $10 million.
Error Location:	`Amount`

Opportunity Check for Products

Field	Value
Description:	Validates that an opportunity has at least one opportunity product before users can save a change to an opportunity.
Formula:	`NOT(OR(ISNEW(),HasOpportunityLineItem))`
Error Message:	You must add products to this opportunity before saving.
Error Location:	`Top of Page`

Opportunity Must Have Products if Beyond "Needs Analysis" Stage

Field	Value
Description:	Validates that an opportunity has opportunity products before the `Stage` can move beyond Needs Analysis.
Formula:	`AND ( CASE( StageName, "Value Proposition", 1, "Id. Decision Makers", 1, "Perception Analysis", 1, "Proposal/Price Quote", 1, "Negotiation/Review", 1, "Closed Won", 1, 0) = 1, NOT(HasOpportunityLineItem) )`
Error Message:	Opportunity products are required to advance beyond the Needs Analysis stage.
Error Location:	`Top of Page`

Opportunity Name Format

Field	Value
Description:	Validates that an opportunity contains a hyphen as a way of enforcing an "[Account] - [Amount]" opportunity naming convention.
Formula:	`FIND( " - ", Name ) = 0`
Error Message:	Opportunity Name should use "[Account] - [Amount]" format.
Error Location:	`Opportunity Name`

Prevent Sales Reps from Moving Opportunity Stage Backwards

Field	Value
Description:	Prevent sales reps from changing opportunity `Stage` "backwards" to specific values, once they have accepted the opportunity via a custom approval process. The approval process sets the custom `Accepted Flag` checkbox to True.
Formula:	`AND( Accepted_Flag__c, OR ( ISPICKVAL( StageName, "Stage 1"), ISPICKVAL( StageName, "Stage 2")) )`
Error Message:	Invalid stage for accepted opportunity.
Error Location:	`Stage`

Probability Must Be 100% for Won Opportunities

Field	Value
Description:	Validates that the probability of a won opportunity is properly set to 100%. This is useful for data cleanliness and reporting purposes.
Formula:	`AND ( ISPICKVAL( StageName, "Closed Won"), Probability <> 1)`
Error Message:	Probability must be 100% for won opportunities.
Error Location:	`Probability`

Probability Must Be Zero for Lost Opportunities

Field	Value
Description:	Validates that the probability of a lost opportunity is properly set to zero. This is useful for data cleanliness and reporting purposes.
Formula:	`AND ( ISPICKVAL( StageName, "Closed Lost"), Probability <> 0)`
Error Message:	Probability must be 0% for lost opportunities.
Error Location:	`Probability`

Project Start Date

Field	Value
Description:	Validates that a field is conditionally required based on the values of other fields. Use this validation formula to ensure that users include a `Project Start Date` for an opportunity that is closed/won.
Formula:	`AND(ISPICKVAL(StageName, "Closed Won"), ISNULL(Project_Start_Date__c))`
Error Message:	Project start date is required for won opportunities.
Error Location:	`Project Start Date`

Sample Quote Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Display Error if Quote Line Item Discount Exceeds 40%

Field	Value
Description:	Shows an error if a quote line item's discount exceeds 40%.
Formula:	`Discount > .40`
Error Message:	The discount on this quote line item cannot exceed 40%.
Error Location:	`Discount` on quote

Sample User, Role, and Profile Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Discount Percent Does Not Exceed Role-Based Limit

Field	Value
Description:	Validates that a custom field on opportunities called `Discount Percent` does not exceed a maximum value that varies depending on the user's role. The default maximum is 15%.
Formula:	`Discount_Percent__c > VLOOKUP($ObjectType.Role_Limits__c.Fields.Limit__c, $ObjectType.Role_Limits__c.Fields.Name, $UserRole.Name)`
Error Message:	Discount (%) exceeds limit allowed for your role.
Error Location:	`Discount Percent`

Expense Amount Does Not Exceed User's Max Allowed Expense

Field	Value
Description:	Validates a custom field called `Expense Amount` against a custom user field called `Max Allowed Expense`.
Formula:	`Expense_Amount__c > $User.Max_Allowed_Expense__c`
Error Message:	Amount cannot exceed your maximum allowed expense.
Error Location:	`Expense Amount`

Only Record Owner Can Change Field

Field	Value
Description:	Ensures that only the record owner can make changes to a custom field called `Personal Goal`.
Formula:	`AND( ISCHANGED( Personal_Goal__c ), Owner <> $User.Id )`
Error Message:	Only record owner can change Personal Goal.
Error Location:	`Personal Goal`

Only Record Owner or Administrator Can Change Field

Field	Value
Description:	Ensures that a user can make changes to a custom field called `Personal Goal` only if the user is the record owner or has a custom profile of "Custom: System Admin."
Formula:	`AND( ISCHANGED( Personal_Goal__c ), Owner <> $User.Id, $Profile.Name <> "Custom: System Admin" )` Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Only record owner or administrator can change Personal Goal.
Error Location:	`Personal Goal`

Opportunity Close Date Can Only Be Back-Dated by Administrator

Field	Value
Description:	Validates that the `Close Date` of an opportunity does not fall prior to the current month, except for users who have a custom profile called "Custom: System Admin."
Formula:	`AND( OR ( ISNEW(), ISCHANGED( CloseDate )), CloseDate < DATE( YEAR(TODAY()), MONTH(TODAY()), 1), $Profile.Name <> "Custom: System Admin" )` Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Close Date cannot be prior to current month.
Error Location:	`Close Date`

Miscellaneous Sample Validation Rules

For more information on any of the formula functions used in these examples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Allow Number to Be Increased but Not Decreased

Field	Value
Description:	Allows a custom field called `Commit Amount` to be increased but not decreased after initial creation. This rule uses the PRIORVALUE() function to compare the updated value of the field to its value prior to update.
Formula:	`PRIORVALUE(Commit_Amount__c) > Commit_Amount__c`
Error Message:	Commit Amount cannot be decreased.
Error Location:	`Commit Amount`

California Driver's License

Field	Value
Description:	Ensures that a custom field called `Drivers License` is in the correct A9999999 format when the `Mailing State` is "CA".

Formula:	`AND(`
	`MailingState = "CA",`
	`NOT(REGEX(Drivers_License__c, "([A-Z]\\d{7})?"))`
	`)`

Error Message:	Invalid California driver's license format.
Error Location:	`Drivers License`

Force Users to Check "I Accept Terms" to Enter Certain Values

Field	Value
Description:	Uses a checkbox labeled "I accept terms" to force the user to select a checkbox in order to enter a value called `Number of Days` that exceeds their Paid Time Off (PTO) balance available.
Formula:	`AND( NOT( I_accept_terms__c ), Number_of_Days__c > $User.PTO_Balance__c )`
Error Message:	Request will cause a negative PTO balance. You must accept Negative PTO Balance terms.
Error Location:	`I accept terms`

Prohibit Changes to a Field After It Has Been Saved

Field	Value
Description:	Prevents users from changing a custom field called `Guaranteed Rate` after it has been saved initially.
Formula:	`AND( NOT( ISNEW() ), ISCHANGED( Guaranteed_Rate__c ) )`
Error Message:	Guaranteed Rate cannot be changed.
Error Location:	`Guaranteed Rate`

Social Security Number Format

Field	Value
Description:	Validates that a custom text field called `SSN` is formatted in 999-99-9999 number format (if it is not blank). The pattern specifies: • Three single digits (0-9):\\d{3} • A dash • Two single digits (0-9):\\d{2} • A dash • Four single digits (0-9):\\d{4}

Formula:	`NOT(`
	`OR(`
	`ISBLANK(Social_Security_Number__c),`
	`REGEX( Social_Security_Number__c , "[0-9]{3}-[0-9]{2}-[0-9]{4}")`
	`)`
	`)`

Error Message:	SSN must be in this format: 999-99-9999.
Error Location:	`SSN`

Valid Currency

Field	Value
Description:	Validates selected currency against an explicit subset of active currencies in your organization using the `Currency` picklist. Use this example if you only allow some of the active currencies in your organization to be applied to certain types of records.

Formula:	`CASE(CurrencyIsoCode,`
	`"USD", 1,`
	`"EUR", 1,`
	`"GBP", 1,`
	`"JPY", 1,`
	`0) = 0`

Error Message:	Currency must be USD, EUR, GBP, or JPY.
Error Location:	`Currency`

Valid Credit Card Number

Field	Value
Description:	Validates that a custom text field called `Credit_Card_Number` is formatted in 9999-9999-9999-9999 or 9999999999999999 number format when it is not blank. The pattern specifies: • Four digits (0-9) followed by a dash: \\d{4}- • The aforementioned pattern is repeated three times by wrapping it in () {3} • Four digits (0-9) • The OR character (|) allows an alternative pattern of 16 digits of zero through nine with no dashes: \\d{16}
Formula:	`NOT( REGEX( Credit_Card_Number__c , "(((\\d{4}-){3}\\d{4})|\\d{16})?"))`
Error Message:	Credit Card Number must be in this format: 9999-9999-9999-9999 or 9999999999999999.
Error Location:	`Credit Card Number`

Valid IP Address

Field	Value
Description:	Ensures that a custom field called `IP Address` is in the correct format, four 3-digit numbers (0-255) separated by periods.
Formula:	`NOT( REGEX( IP_Address__c, "^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$" ))`
Error Message:	Error: IP Address must be in form 999.999.999.999 where each part is between 0 and 255.
Error Location:	`IP Address`

Website Extension

Field	Value
Description:	Validates a custom field called `Web Site` to ensure its last four characters are in an explicit set of valid website extensions.

Formula:	`AND(`
	`  RIGHT( Web_Site__c, 4) <> ".COM",`
	`  RIGHT( Web_Site__c, 4) <> ".com",`
	`  RIGHT( Web_Site__c, 4) <> ".ORG",`
	`  RIGHT( Web_Site__c, 4) <> ".org",`
	`  RIGHT( Web_Site__c, 4) <> ".NET",`
	`  RIGHT( Web_Site__c, 4) <> ".net",`
	`  RIGHT( Web_Site__c, 6) <> ".CO.UK",`
	`  RIGHT( Web_Site__c, 6) <> ".co.uk"`
	`)`

Error Message:	Web Site must have an extension of `.com`, `.org`, `.net`, or `.co.uk`.
Error Location:	`Web Site`

Require Field Input to Ensure Data Quality

Improve the quality of data that users enter in Salesforce by creating universally required fields.

A universally required field is a custom field that must have a value whenever a record is saved within Salesforce, the Force.com API, Connect Offline, Salesforce for Outlook, the Self-Service portal, or automated processes such as Web-to-Lead and Web-to-Case. Making a field required on a page layout or through field-level security ensures users must enter a value, but making a field required universally gives you a higher level of data quality beyond the presentation level of page layouts.

You can make the following types of custom fields universally required:

- Currency
- Date
- Date/Time
- Email
- Master-Detail Relationship (always required)
- Number
- Percent
- Phone
- Text
- Text Area
- URL

To make a custom field universally required, select the **Required** checkbox when defining the custom field.

Note: You must specify a default value for required campaign member custom fields.

If you make a user field universally required, you must specify a default value for that field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Connect Offline, Salesforce for Outlook, the Self-Service portal, Web-to-Lead, and Web-to-Case are not available in **Database.com**

.

Relationship group members do not support universally required fields.

SEE ALSO:

Considerations for Universally Required Fields

Considerations for Universally Required Fields

A universally required field is a custom field that must have a value whenever a record is saved within Salesforce, the Force.com API, Connect Offline, Salesforce for Outlook, the Self-Service portal, or automated processes such as Web-to-Lead and Web-to-Case. Review the following considerations before making your custom fields universally required.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Standard Objects, Page Layouts, Connect Offline, Salesforce for Outlook, the Self-Service portal, Web-to-Lead, and Web-to-Cases are not available in **Database.com**

- Standard fields cannot be universally required.
- Universally required fields are required across all record types.
- Edit pages always display universally required fields, regardless of field-level security.
- When designing your page layouts, universally required fields:
  - Cannot be removed from a page layout
  - Are automatically added to the end of the first section of a page layout if not already on it
  - Cannot be read only or optional
  - Display in bold, indicating they are always visible
  - Are disabled on the field properties page because you cannot remove the required setting
- Universally required fields are only enforced during lead conversion if validation and triggers for lead conversion are enabled in your organization.
- Quick Create does not enforce universally required fields.
- If you make an activity custom field universally required, you must also provide a default value.
- You must include universally required fields in your import files or the import will fail.
- Don't assign default values to fields that are both required and unique, because uniqueness errors can result.
- You cannot make a field universally required if it is used by a field update that sets the field to a blank value.
- Required fields may be blank on records that existed before making the field required. When a user updates a record with a blank required field, the user must enter a value in the required field before saving the record.
- Web-to-Lead and Web-to-Case request data is not validated by Salesforce. Invalid data isn't saved when requests are submitted. For example, if your custom field is a currency field and a user enters alphabetic characters such as "Abc" instead of numbers, the request is still submitted but with no value saved in the custom currency field.

SEE ALSO:

Require Field Input to Ensure Data Quality

About Field Sets

A field set is a grouping of fields. For example, you could have a field set that contains fields describing a user's first name, middle name, last name, and business title. When a field set is added to a Visualforce page, developers can loop over its fields and render them. If the page is added to a managed package, administrators can add, remove, or reorder fields in a field set to modify the fields presented on the Visualforce page without modifying any code. The same Visualforce page can present different sets of information, depending on which fields a subscriber prefers to keep.

As an administrator, you can create or edit field sets for your organization, or edit any installed field set. Field sets are available on all standard objects that support custom fields, and any organization that supports creating Visualforce pages.

EDITIONS

Available in: Salesforce Classic

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Fields added to a field set can be in one of two categories:

- If a field is marked as `Available for the Field Set`, it exists in the field set, but the developer hasn't presented it on the packaged Visualforce page. Administrators can display the field after the field set is deployed by moving it from the `Available` column to the `In the Field Set` column.
- If a field is marked as `In the Field Set`, the developer has rendered the field on the packaged Visualforce page by default. Administrators can remove the field from the page after the field set is deployed by removing it from the `In the Field Set` column.

SEE ALSO:

Field Sets Required Bit

Create Custom Fields

Developer's Guide: Visualforce Developer's Guide

ISVforce Guide

Creating and Editing Field Sets

Salesforce has a drag-and-drop WYSIWYG tool for creating and editing field sets The enhanced field sets editor is enabled by default, and provides all of the functionality of the original editor, as well as additional functionality and an easier-to-use WYSIWYG interface.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

1. From the management settings for the appropriate object, go to Field Sets, and then click **New**.
2. Enter a **Field Set Label**. This is the name presented to subscribers who install the field through a managed package.
3. Optionally, enter a name for your field set. This is used by your Visualforce page to reference the field set.
4. In the **Where is this used?** area, provide a brief description of which Visualforce pages use the field set, and for what purpose. This information helps a subscriber understand where and how an installed field set is being used, so that they can populate it with their own fields.
5. Save your changes.
6. To add fields to the field set, drag the fields from the object palette and drop them into the `Available for the Field Set` or the `In the Field Set` container. The fields in the `Available for the Field Set` container are not initially visible on the Visualforce page. The fields in the `In the Field Set` container are visible by default.

   Note: In the field set, you can span to fields that reference multiple objects. When you span a field into a field set that references multiple objects, the only field you can span to is the `Name` object.

You can drag and drop a field from one container to the other. The vertical order of the `In the Field Set` list indicates the order of how the fields render on Visualforce pages.

7. To remove a field from the field set, drag the element back to the object palette, or click the icon next to the element.
8. To make a field required, double click the element or click the wrench icon ( ) next to it and select the `Required` checkbox.

   Note: * Indicates the field is required and must have a value to save the record.

9. Save your changes.

Important: The total number of cross object spans within the `In the Field Set` container can't exceed 25.

After a field set is deployed in your organization, you can always mark fields that are in the `Available for the Field Set` list as `In the Field Set`, or vice versa. To do so:

1. Find the field set that you want to edit. From Setup enter *`Installed Packages`* in the `Quick Find` box, select **Installed Packages**, click an installed package, and then click the field set you want to edit. Alternatively, if you know which object contains the field set you want to edit, go to the object detail page and click **Edit** in the field set related list.
2. If you didn't create the field set initially, you'll only be able to edit the fields within the field set. To move fields between containers, drag and drop a field from one container to the other. To change the order of a rendered field, drag a field up or down the list and drop the field in the order you want it to appear.
3. Save your changes.

Field Sets Required Bit

You can define a field as required when you create or edit field sets. You may want to define a field as required to ensure a user enters the necessary information on a field. The required field is only available in the `In the Field Set` container. If you define a field as required in the `In the Field Set` container, and remove the field from the `In the Field Set`, the required attribute is removed.

Note: If you remove fields that were made required by an installed managed package from the `In the Field Set` container, the required attribute isn't removed from those fields.

To define a field as required in a field set, see Creating and Editing Field Sets on page 212

EDITIONS

Available in: Salesforce Classic

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

About Field Sets

Creating and Editing Field Sets

Roll-Up Summary Field

A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record. The detail record must be related to the master through a master-detail relationship. For example, you want to display the sum of invoice amounts for all related invoice custom object records in an account's Invoices related list. You can display this total in a custom account field called Total Invoice Amount.

You can perform different types of calculations with a roll-up summary field. You can count the number of detail records related to a master record. Or, you can calculate the sum, minimum value, or maximum value of a field in the detail records.

Before you begin creating roll-up summary fields for your organization, review the implementation tips and best practices.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Implementation Tips

Administration

- Create roll-up summary fields on:
  - Any custom object that is on the master side of a master-detail relationship
  - Any standard object that is on the master side of a master-detail relationship with a custom object
  - Opportunities using the values of opportunity products related to the opportunity
  - Accounts using the values of related opportunities
  - Campaigns using campaign member status or the values of campaign member custom fields

    Note: Campaign member custom formula fields that reference fields derived from leads or contacts are not supported.

- The types of fields you can calculate in a roll-up summary field depend on the type of calculation. For example,
  - Number, currency, and percent fields are available when you select *SUM* as the roll-up type.
  - Number, currency, percent, date, and date/time fields are available when you select *MIN* or *MAX* as the roll-up type.
- Sometimes you can't change the field type of a field that you reference in a roll-up summary field.
- Make sure that the filter for your roll-up summary doesn't encounter a formula field that results in "#Error!". If one of your filter criteria uses a formula field that results in an error, no matches are returned for that filter criterion. For example, your roll-up summary filter is "Formula Field equals 10". Two records contain errors, and one contains the value "10" in that field. In this case, your summary includes only the record with the value "10."
- When you delete a child record on a roll-up Summary field, Salesforce doesn't recalculate the value of the field. Select the `Force a mass recalculation on this field` option on the edit page of the roll-up summary field to manually recalculate the value.
- You can't use long text area, multi-select picklist, `Description` fields, system fields like `Last Activity`, cross-object formula fields, and lookup fields in the field column of roll-up summary filters.
- Auto number fields are not available as the field to aggregate in a roll-up summary field.
- After you have created a roll-up summary field on an object, you cannot convert the object's master-detail relationship into a lookup relationship.
- Roll-up summary fields are not available for mapping lead fields of converted leads.

Management

- If a roll-up summary field doesn't contain cross-object field references or functions that derive values on the fly, such as NOW or TODAY, it can calculate the values of formula fields.

  Note: The value of a formula field can result in "#Error!", which affects the summarized total. If your roll-up summary type is COUNT, records are included regardless of whether they contain a formula field with an error. However, when the `Field to Aggregate` is a formula field that results in "#Error!", calculations of type MIN, MAX, and SUM exclude those formula values.

- Changes to the value of a roll-up summary field can trigger assignment rules to run. If a roll-up summary field is part of the criteria in an assignment rule, the field's new value is used to evaluate whether to reassign the record.
- A roll-up summary field can trigger workflow rules and field validations. However, workflow rules and field validations do not fire when the following changes cause a mass recalculation of roll-up summary values:
  - Changing the roll-up summary definition (such as the object, function, or field being aggregated)
  - Changing the expression of a formula field referenced in a roll-up summary field
  - Replacing picklist values for picklist fields referenced in the roll-up summary filter
  - Changing picklist record type definitions
  - Changing currency conversion rates
  - Changing price book entries
- Calculating roll-up summary field values can take up to 30 minutes, depending on the number of records affected and other factors.
- You aren't prevented from creating roll-up summary fields that can result in invalid values, such as February 29 in a non-leap year. If a roll-up summary field results in an invalid value, the value is not recalculated. The field continues to display with an invalid roll-up summary icon ( ) until you change the values being summarized.
- If your organization uses multiple currencies, the currency of the master record determines the currency of the roll-up summary field. For example, if the master and detail records are in different currencies, the detail record value is converted into the currency of the master record.
- Changing a conversion rate triggers roll-up summary fields to recalculate. If you're using multiple currencies, we recommend changing the conversion rate from Manage Currencies in Setup, and not from the API. If you change the rate from the API, related jobs that are less than 24 hours old can interfere with your change. For details, see Edit Conversion Rates.
- If your organization has advanced currency management enabled, currency roll-up summary fields are invalid if they are on accounts and summarizing opportunity values, or on opportunities and summarizing custom object values.
- Salesforce prevents users from saving a record if it invalidates a related record. For example, a master record has a validation rule that requires the roll-up summary field value to be greater than 100. If the user's change to a related child record would put the value over 100, the user can't save the record.
- If a lookup field references a record that has been deleted, Salesforce clears the value of the lookup field by default. Alternatively, you can choose to prevent records from being deleted if they're in a lookup relationship.

  To be used in a roll-up summary field with a `Roll-Up Type` of `COUNT` or `SUM`, the lookup field must have the `What to do if the lookup record is deleted?` option set to *`Don't allow deletion of the lookup record that's part of a lookup relationship`*. If the option *`Clear the value of this field. You can't choose this option if you make the field required`* is selected, you can't create a `COUNT` or `SUM` roll-up summary field that pulls data from your lookup field.

Best Practices

- Apply field-level security to your roll-up summary fields if they calculate values that you do not want visible to users. Fields that your users cannot see due to field-level security settings on the detail record are still calculated in a roll-up summary field.

- If you have validation rules, consider how they affect roll-up summary fields. The value in a roll-up summary field changes when the values in the detail records change. So, validation errors can display when saving either the detail or master record.
- Because roll-up summary fields are not displayed on edit pages, you can use them in validation rules but not as the error location for your validation.
- Avoid referencing a roll-up summary field from a child record. The roll-up summary fields referenced from child records can have outdated values, because their parent records have not been updated. Instead, reference roll-up summary fields from parent records. Your roll-up summary fields will always have updated values, because that rule runs after the parent value has been updated.

  If you're trying to enforce a record limit of 25 on the parent roll-up summary field, create validation rules on your child objects. When you add a child record, your validation rule on the child object can check if the count is already 25 or greater.

  ```
  AND(ISNEW(), Sample.Line_Count__c >= 25)
  ```

- Plan your implementation of roll-up summary fields carefully before creating them. Once created, you cannot change the detail object selected or delete any field referenced in your roll-up summary definition.
- Advanced currency management affects roll-up summary fields. If your organization enables advanced currency management, delete the currency roll-up summary fields on accounts that summarize opportunity values and on opportunities that summarize custom object values. Otherwise, the fields continue to display with an invalid roll-up summary icon because their values are no longer calculated.
- Automatically derived fields, such as current date or current user, aren't allowed in a roll-up summary field. Forbidden fields include formula fields containing functions that derive values on the fly, such as DATEVALUE, NOW, and TODAY. Formula fields that include related object merge fields are also not allowed in roll-up summary fields.
- When you refer to a roll-up summary field in a list view or report, you can't use certain qualifiers, including:
  - **Starts with**
  - **Contains**
  - **Does not contain**
  - **Includes**
  - **Excludes**
  - **Within**

SEE ALSO:

Defining Roll-Up Summaries

Create Custom Fields

Defining Roll-Up Summaries

Define roll-up summary fields on the object that is on the master side of a master-detail relationship. If a relationship does not already exist, first create a master-detail relationship between the master object that displays the value and the detail object containing the records you are summarizing.

To define a roll-up summary field:

1. Create a custom field on the object where you want the field displayed. Summary fields summarize the values from records on a related object, so the object on which you create the field should be on the master side of a master-detail relationship. For instructions on creating a custom field, see Create Custom Fields on page 125.
2. Choose the `Roll-Up Summary` field type, and click **Next**.
3. Enter a field label and any other attributes. Click **Next**.
4. Select the object on the detail side of a master-detail relationship. This object contains the records you want to summarize.
5. Select the type of summary:

Type	Description
COUNT	Totals the number of related records.
SUM	Totals the values in the field you select in the `Field to Aggregate` option. Only number, currency, and percent fields are available.
MIN	Displays the lowest value of the field you select in the `Field to Aggregate` option for all directly related records. Only number, currency, percent, date, and date/time fields are available.
MAX	Displays the highest value of the field you select in the `Field to Aggregate` option for all directly related records. Only number, currency, percent, date, and date/time fields are available.

6. Enter your filter criteria if you want a selected group of records in your summary calculation. If your organization uses multiple languages, enter filter values in your organization's default language.

   When you use picklists to specify filter criteria, the selected values are stored in the organization's default language. If you edit or clone existing filter criteria, first set the `Default Language` on the Company Information page to the same language that was used to set the original filter criteria. Otherwise, the filter criteria may not be evaluated as expected.

7. Click **Next**.
8. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
9. Choose the page layouts that should display the field. The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is automatically added to the bottom of the user detail page.
10. Click **Save** to finish or **Save & New** to create more custom fields.

SEE ALSO:

Roll-Up Summary Field

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

USER PERMISSIONS

To view roll-up summary field definitions:

- "View Setup and Configuration"

To edit roll-up summary field definitions:

- "Customize Application"

Lookup Filters

Improve user productivity and data quality with lookup filters. Lookup filters are administrator settings that restrict the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

Administrators specify the restrictions by configuring filter criteria that compare fields and values on:

- The current record (source)
- The lookup object (target)
- The user's record, permissions, and role
- Records directly related to the target object

For example, you can:

- Restrict the `Account Name` field on opportunities to allow only accounts with a record type of Customer, filtering out Partner and Competitor.
- Restrict the `Account Name` field on opportunities to allow only active accounts.
- Restrict the `Contact` field on cases to allow only contacts associated with the account specified in the `Account Name` field on the case record.
- Restrict the `Account Name` field on cases to allow only users with the "International Sales" profile to create or edit cases for accounts outside the United States.

Tip: When you define a lookup filter, optionally click `Insert Suggested Criteria` to choose from a list of lookup filter criteria that Salesforce suggests based on the relationships between objects in your org.

Administrators can make lookup filters required or optional.

- For fields with required lookup filters, only values that match the lookup filter criteria appear in the lookup dialog. Also, users can't save invalid values manually typed in the field when editing the record. If a user tries to save an invalid value, Salesforce displays an error message, which administrators can customize.
- For fields with optional lookup filters, only values that match the lookup filter criteria appear in the lookup dialog initially; however, users can click the **Show all results** link in the lookup dialog to remove the filter and view all search result values for the lookup field. Also, optional lookup filters allow users to save values that don't match the lookup filter criteria, and Salesforce does not display an error message if a user tries to save such values.

Lookup filter criteria can compare fields on the source object with different types of fields on the target object as long as the fields are compatible.

Source Object Field Type	Compatible Target Object Field Types
Currency	Currency, Roll-Up Summary
Date	Date, Date/Time, Roll-Up Summary
Date/Time	Date, Date/Time, Roll-Up Summary
Hierarchy	Hierarchy, Lookup, Master-Detail
Lookup	Hierarchy, Lookup, Master-Detail
Master-Detail	Lookup, Hierarchy, Master-Detail
Number	Number, Percent, Roll-Up Summary

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To manage lookup filters:
- "Customize Application"

Percent	Number, Percent, Roll-Up Summary
Picklist	Text, Text Area, Email, URL
Roll-Up Summary	Currency, Number, Date, Date/Time, Roll-Up Summary

Supported Objects

Salesforce supports lookup filters on relationship fields that point to:

- Accounts
- Assets
- Badges
- Badges Received
- Business Hours
- Campaigns
- Cases
- Contacts
- Content Folders
- Contracts
- Endorsements
- Entitlements
- Ideas
- Leads
- Opportunities
- Order Products
- Orders
- Price Books
- Products
- Quotes
- Service contracts
- Skill Users
- Skills
- Social Personas
- Solutions
- Thanks
- User Provisioning Accounts
- User Provisioning Logs
- User Provisioning Requests
- Users
- Work Order Line Items
- Work Orders

- Zones
- Custom objects

SEE ALSO:

Notes on Lookup Filters

Lookup Filter Examples

Define Lookup Filters

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To define lookup filters:
- "Customize Application"

1. You can create lookup filters for new relationship fields in step 3 of the custom field wizard.
   a. From the management settings for the field's object, go to Fields.
   b. Click **Edit** next to the name of the lookup or master-detail relationship field to which you want to apply the filter.
2. In the Lookup Filter Options section, click **Show Filter Settings**.
3. Specify the filter criteria a record must meet to be a valid value. To specify criteria, click **Insert Suggested Criteria** and choose from a list of suggested criteria, or manually enter your own criteria. To enter your own criteria:
   a. In the first column, click the lookup icon or start typing in the text box and select a field.
   b. In the second column, select an operator.
   c. In the third column, select `Value` if Salesforce should compare the field in the first column with a static value, or select `Field` if Salesforce should compare the field in the first column with the value of another field.
   d. In the fourth column, enter the value or select the field that Salesforce should compare with the field in the first column.

   Note:
   - Click **Add Filter Logic** to add Boolean conditions.
   - Select a suggested field from the Field text box. You can only select fields on the current record, the lookup object, or the user record. You can also choose related fields that are one relationship away from the lookup object. Salesforce assists you by listing the available fields and relationships when you click the lookup icon or click inside the text box.
   - Lookup filter criteria can compare fields of different types as long as they are compatible.
4. Specify whether the filter is required or optional. For fields with optional lookup filters, only values that match the lookup filter criteria appear in the lookup dialog initially; however, users can click the **Show all results** link in the lookup dialog to remove the filter and view all search result values for the lookup field. Also, optional lookup filters allow users to save values that don't match the lookup filter criteria, and Salesforce does not display an error message if a user tries to save such values.

   For required lookup filters, specify whether you want Salesforce to display the standard error message or a custom message when a user enters an invalid value.
5. Optionally, enter text to display in the lookup search dialog. Consider text that guides users in their searches and explains the business rule that the lookup filter implements.
6. Leave `Enable this filter` selected.

7. Save your changes.

SEE ALSO:

- Notes on Lookup Filters
- Dependent Lookups
- Lookup Filter Examples
- Find Object Management Settings

Delete or Deactivate Lookup Filters

1. From the managements settings for the relationship field's object, go to Fields.
2. Scroll to the Custom Fields & Relationships related list.
3. Click the name of the field containing the lookup filter.
4. Click **Edit**.
5. To deactivate the lookup filter, deselect `Enable this filter`, then save your changes.

   Deactivating a lookup filter preserves the lookup filter configuration but:

   - Prevents it from applying to the relationship field
   - Prevents it from impacting the cross-object references limit
   - Removes it as a dependency for fields referenced in the lookup filter criteria
6. To delete the lookup filter, click **Clear Filter Criteria**, then save your changes.

   Deleting a lookup filter permanently removes it. You can't recover deleted lookup filters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To define lookup filters:

- "Customize Application"

SEE ALSO:

- Dependent Lookups
- Notes on Lookup Filters
- Find Object Management Settings

View a List of Lookup Filters for a Target Object

You can quickly see a list of all of the lookup filters that restrict the values of each target object. This is useful when creating similar filters for a target object. Also, lookup filters that reference fields on related objects count against the cross-object reference limit, which is the number of unique relationships allowed for a target object. The Related Lookup Filters list lets you see which lookup filters might impact that limit.

To see which lookup filters affect the limit for a particular target object, from the management settings for the object, go to Related Lookup Filters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To define lookup filters:

- "Customize Application"

SEE ALSO:

- Dependent Lookups
- Notes on Lookup Filters

Dependent Lookups

A *dependent lookup* is a relationship field with a lookup filter that references fields on the source object. For example, you can configure the case `Contact` field to only show contacts associated with the account selected in the case `Account Name` field.

When a user changes the value of a referenced field on the source object, Salesforce immediately verifies that the value in the dependent lookup still meets the lookup filter criteria. If the value doesn't meet the criteria, an error message is displayed and users can't save the record until the value is valid.

If the referenced field on the source object is a lookup, master-detail, or hierarchy field, users can't change its value by typing. Instead, users must click the lookup icon and select a value in the lookup search dialog.

Tip: Dependent lookups are supported in Visualforce pages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To manage dependent lookups:
- "Customize Application"

SEE ALSO:

Define Lookup Filters

Lookup Filter Examples

Notes on Lookup Filters

- On the Fields page, the icon indicates all fields with active lookup filters. The icon indicates that the lookup filter is required.
- The lookup filters you create in Salesforce also appear in the partner portal and Customer Portal.
- Lookup filters are case-sensitive.
- If you convert a required lookup filter with a custom error message to optional, Salesforce deletes the message.
- If you create a lookup filter that invalidates an existing value for that field, the value persists; however, when a user edits the record, Salesforce displays an error message and requires the user to change the invalid value before saving.
- Salesforce prevents you from saving changes that cause required lookup filters on related records to contain invalid values.
- Versions 16.0 and higher of the Salesforce API support lookup filters. Lookup filters are enforced when you load data through the API.
- If you configure a lookup filter to show inactive users only, the relationship field has no valid options because inactive users are never valid for relationship fields that point to the User object.
- If you create a filtered lookup on a field that looks up to another object, both objects must be deployed into the organization at the same time.
- Salesforce does not display an error message if the value of a controlling field invalidates the value of a dependent master-detail relationship field.
- Dependent lookups are supported in Visualforce pages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS

To manage lookup filters:
- "Customize Application"

Spanning Relationships in Lookup Filters

Filter criteria can include fields directly related to the target object (one level only). For example, on a lookup field pointing to contacts, a lookup filter can reference fields on the account related to the contact via the `Account Name` relationship field. The lookup field can also reference fields on the contact related to the contact via the `Reports To` relationship field.

For required lookup filters, each field referenced on a related lookup object counts against the number of unique relationships allowed for the referenced object, not the source object. For example, the two unique relationships described above count against the number allowed for the Contact object. Optional lookup filters don't count against the limit on the number of unique relationships allowed per object.

To see which lookup filters affect the limit for a particular target object, from the management settings for the object, go to Related Lookup Filters.

Lookup Filters vs. Validation Rules

Validation rules and lookup filters achieve similar ends, but offer different advantages. Use a lookup filter if:

- You want to improve user efficiency by limiting the number of available options in a lookup search dialog.
- You want to improve user efficiency by automating filters on lookup search dialogs that your users manually set.

Use a validation rule if:

- You're close to the maximum number of lookup filters allowed.
- You must implement a complex business rule that requires you to use a formula. Formulas can reference fields that basic filter criteria can't reference, such as fields on the parent of the source object. Formulas can also use functions. For example, use `ISNEW` if the rule should only apply on record creation, or `ISCHANGED` if the rule should apply when a field changes.

SEE ALSO:

Lookup Filters

Dependent Lookups

Notes on Using Lookup Filters with Person Accounts

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

If your organization uses person accounts, note the following:

- Person Accounts don't support Contact filters; however, Person Accounts support Account filters. For example, if the `Account` field has a dependent lookup filter that's added to a Person Account, dependent lookups are supported. If the `Contact` field has a dependent lookup filter that's added to a Person Account, dependent lookups isn't supported.
- Lookup filter criteria on `Account Name` only apply to business accounts, not person accounts. For example, if your lookup filter criteria is `Account Name does not contain book`, business accounts with "book" in the name, such as John's Bookstore, are not valid, but person accounts with "book" in the name, such as John Booker, are valid and appear in the lookup dialog for the `Account` field. If you need to filter on the name for a person account, use the `First Name` or `Last Name` fields instead.
- Use the `Is Person Account` field in your lookup filter criteria to restrict the valid values of a lookup field to one type of account (either person accounts or business accounts). For example, to restrict a lookup to only person accounts, include the following in your lookup filter criteria: *`Is Person Account equals True`*.

- You can't package lookup filters that reference standard fields specific to person accounts, such as the `Email` and `Title` fields.

SEE ALSO:

Lookup Filters

Lookup Filters Best Practices

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

Custom Help

Define custom help for fields with lookup filters to let users know about the business rule the filter enforces. For example, if the lookup filter restricts the `Account Name` on opportunities to only allow active accounts, define custom help that states *`You can only associate active accounts with opportunities.`*

Error Messages

Customize lookup filter error messages to guide users who type invalid values. For example, if the lookup filter restricts the `Account Name` on opportunities to only allow active accounts, define an error message that states *`Value doesn't exist or isn't an active account.`*

Important: Salesforce translates the standard error message for required lookup filters, but not custom error messages. Use the Translation Workbench to translate lookup filter custom error messages. To restore the standard error message after modifying it, click **Reset to default message**.

Working with Master-Detail Relationship Fields

When creating a lookup filter on a master-detail relationship field, verify that the current values of the field on all of the detail records meet the criteria you specify. If you specify criteria that an existing value doesn't meet, Salesforce prevents the user from saving changes to the detail record. If this occurs, the user must first modify the value on the master record to meet the criteria. For example, consider a custom object with a master-detail relationship field that points to accounts. If you define a lookup filter that excludes all accounts with a `Create Date` before 01/01/2009, verify that no existing records of that custom object have a master-detail relationship with any account created before 2009. A quick way to do this is to create a report that shows all accounts with a `Create Date` before 01/01/2009.

Profile-Based Lookup Filters

Use *`Current User Profile: ID`* in filter criteria to define different filter criteria for different users, or to let administrators enter values that don't match the criteria. Avoid using *`Current User Profile: Name`* due to technical limitations on standard profiles.

If you enter *`Current User Profile: Name`* or *`Profile: Name`* in the Field column of your lookup filter criteria, Salesforce displays a lookup icon in that row. Click the lookup icon to select from a list of existing profiles rather than typing profile names.

Record IDs vs. Record Names

To reference a specific record in filter criteria, use the ID of the record instead of its name. IDs are always unique whereas names are not.

Testing

After creating a lookup filter, test it to make sure it is not too restrictive. Depending on their permissions, some users may have read-only access to some relationship fields; ensure your lookup filters don't prevent those users from editing records critical to their job functions.

Dependent Lookups on Page Layouts and Mini Page Layouts in the Console

When designing page layouts with dependent lookups:

- If a dependent lookup is above its controlling field on a layout, make its lookup filter optional or redesign the layout. Moving a required dependent lookup above its controlling field may confuse users who typically start from the top of a page when entering data.
- Ensure that both the controlling and dependent fields are visible so users can correct invalid values.

Lookup Filters and the Lookup Filter Fields Search Layout

Don't reference the same fields in both lookup filter criteria and the Lookup Filter Fields search layout. Users might assume that results from their custom search override administrator-controlled lookup filters.

SEE ALSO:

Lookup Filters

Lookup Filter Examples

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To define lookup filters:
- "Customize Application"

Record Types in Lookup Filters

If the value of a relationship field should only consist of records with a particular record type, specify the record type in a lookup filter. For example, if the `Account Name` field on opportunities should only have accounts with a Customer Account custom record type, define the following lookup filter to restrict users to only creating or editing opportunities associated with accounts that have a Customer Account record type, excluding accounts with Partner Account and Competitor Account record types:

Filter Criteria

Account Name: Account Record Type equals value Customer Account

Custom Error Message

Account does not exist or is not a customer account.

Lookup Window Text

You can only associate customer accounts to an opportunity. Search results only display customer accounts.

Record Status in Lookup Filters

If the value of a relationship field should only consist of records with particular status, specify the status in a lookup filter. For example, consider a Job Application object with a relationship field that points to the `Position` object. If the relationship field should only have open positions, define the following lookup filter to restrict users to only creating or editing job applications for positions with the `Status` field set to Open:

Filter Criteria

Position: Status equals value Open

Custom Error Message

Position does not exist or is not an open position.

Lookup Window Text

You can associate only open positions with job applications. Search results display open positions only.

Profiles in Lookup Filters

When a business rule does not apply to users with every profile, use the `Current User Profile` global variable fields to define lookup filters that only affect users with a particular profile.

For example, the following lookup filter on the Case object `Account Name` field restricts users with a "Domestic Sales" profile to only creating or editing opportunities associated with accounts that have a billing country of "USA" while allowing other users to associate opportunities with any account:

Filter Criteria

1. *Current User Profile: Name equals value Domestic Sales*
2. *Account Name: Billing Country equals value USA*
3. *Current User Profile: Name not equal to value Domestic Sales*

Filter Logic

`(1 AND 2) OR 3`

Custom Error Message

Account does not exist or the account billing country is not USA. Domestic sales reps can only create opportunities for accounts in the United States.

Lookup Window Text

Search results show only United States accounts in the for domestic sales representatives.

You can modify the above example to simultaneously restrict users with a "Global Sales" custom profile to only associating opportunities to accounts with a non-US billing country:

Filter Criteria

1. *Current User Profile: Name equals value Global Sales*
2. *Account Name: Billing Country not equal to value USA*
3. *Current User Profile: Name equals value Domestic Sales*
4. *Account Name: Billing Country equals value USA*
5. *Current User Profile: Name not equal to value Global Sales, Domestic Sales*

Filter Logic

`(1 AND 2) OR (3 AND 4) OR 5`

Custom Error Message

Account does not exist or the account billing country is not in your sales area. Sales reps can only create opportunities for accounts in their sales area.

Lookup Window Text

Search results only display accounts in your region.

Important: If you do not include line 5 in the filter criteria, users who are not in Global Sales or Domestic Sales cannot select or save any values on account records.

Roles in Lookup Filters

When a business rule does not apply to users in every role, use the `Current User Role` global variable fields to define lookup filters that only affect users with particular roles. For example, in a recruiting application that has a Position object with a lookup field to a Compensation Package object, you can restrict users from editing or creating positions that have an executive compensation plan

unless they are executive administrators or vice presidents. To do this, define the following lookup filter on the Position object `Compensation Package Name` field:

Filter Criteria

1. *Current User Role: Name does not start with value VP*
2. *Current User Role: Name does not equal value Executive Administrator*
3. *Compensation Package: Plan Type does not equal value Executive*
4. *Current User Role: Name starts with value VP*
5. *Current User Role: Name equals value Executive Administrator*

Filter Logic

`((1 OR 2) AND 3) OR (4 OR 5)`

Custom Error Message

The compensation plan does not exist, or you have selected an executive compensation plan but do not have access to create executive positions.

Lookup Window Text

Search results only display compensation plans that are relevant to positions you are allowed to create.

Important: Include the condition you are testing and the opposite condition. In this example, lines 1, 2, and 3 of the filter criteria ensure that users who are not VPs or Executive Administrators *cannot* select Executive compensation plans, while lines 4 and 5 ensure that VPs and Executive Administrators *can* select Executive compensation plans.

Blank Values in Lookup Filters

Your lookup filter criteria might reference a field that users often leave blank. You can design your lookup filter criteria to accept blank values by using the **Add Filter Logic** in the filter criteria to create an OR condition. For example, if you have a `Partner Contact` custom field on opportunities, restrict the field to only allow contacts that are associated to an account with a Partner Account record type, or private contacts not associated with any account.

Filter Criteria

1. *Partner Contact: Account: Account Record Type equals value Partner Account*
2. *Partner Contact: Account: Account Name equals value*

Filter Logic

`1 OR 2`

Custom Error Message

The partner contact must be associated with a partner account, or must be a private contact.

Lookup Window Text

Search results only display contacts from partner accounts or your private contacts.

User IDs in Lookup Filters

Using user IDs in optional lookup filters can significantly improve user efficiency by first showing lookup search dialog results that are most relevant to the user while still allowing users to see all results if necessary. For example, on a lookup field to accounts, you can create an optional lookup filter that restricts the search results to accounts that the user owns in the search lookup dialog results. If the user is looking for an account that someone else owns, the user can remove the filter.

Filter Criteria

Current User: User ID equals Field Account: Owner ID

Lookup Window Text

By default, search results only display accounts you own. To search all accounts, click "Show all results."

Simple Dependent Lookups

If the value of a relationship field should depend on the value of another relationship field on the current record, specify the field to field comparison in the criteria. For example, if the case `Contact Name` field should only have contacts associated to the account specified in the case `Account Name` field, use the following lookup filter:

Filter Criteria

Contact Name: Account ID equals field Case: Account ID

Custom Error Message

Contact does not exist or is not associated to the case account.

Lookup Window Text

Search results only display contacts associated to the case account.

Note: When comparing lookup fields in lookup filter criteria, Salesforce always uses the ID of the relationship field, not the name.

Complex Lookup Filters and Dependent Lookups

Achieving complex business rules with lookup filters often involves combing your rules with filter logic and fields of various types. For example, consider an app for booking conference rooms that has the following data model:

Object	Fields
Meeting	• `Meeting Name` • `Office` lookup to the Office object • `Projector Required` checkbox • `Number of Participants` number field • `Conference Room` lookup to the Conference Room object
Conference Room	• `Conference Room Name` • `Has Projector` checkbox • `Number of Seats Available` number field • `Conference Room Location` lookup to the Office object
Office	• `Office Name`

The following lookup filter on the meeting `Conference Room` field restricts the valid values to conference rooms that have a projector if the meeting requires one, as well as the necessary number of seats:

Filter Criteria

1. *Meeting: Projector Required equals field Meeting Conference Room: Has Projector*
2. *Meeting: Projector Required equals value False*
3. *Conference Room: Number of Seats Available greater or equal field Meeting: Number of Participants*

Filter Logic

`(1 OR 2) AND 3`

Custom Error Message

Conference room not found or is insufficient for your meeting.

Lookup Window Text

Search results only display conference rooms that can support your meeting requirements.

To refine the valid values even further, incorporate the office where the conference room is located:

Filter Criteria

1. *Meeting: Projector Required equals field Meeting Conference Room: Has Projector*
2. *Meeting: Projector Required equals value False*
3. *Conference Room: Number of Seats Available greater than field Meeting: Number of Participants*
4. *Meeting: Office equals Field Conference Room: Conference Room Location*

Filter Logic

`(1 OR 2) AND 3 AND 4`

Custom Error Message

Conference room not found or is insufficient for your meeting.

Lookup Window Text

Search results only display conference rooms that can support your meeting requirements.

SEE ALSO:

Notes on Lookup Filters

Formulas

A formula is an algorithm that derives its value from other fields, expressions, or values. Formulas can help you automatically calculate the value of a field based on other fields.

Watch a Demo: Getting Started With Formulas

This video gives a brief introduction to Salesforce formulas, accessing the formulas editor in the app, and how to use the editor tools to create formulas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Reports and Approvals are not available in **Database.com**

IN THIS SECTION:

SEE ALSO:

Elements of a Formula
Formula Data Types

Where are Formulas Used in Salesforce?

Many areas in Salesforce use formulas. Before you begin using formulas, review the differences in their uses.

Use Formulas for:	To:
Approval Processes	Define the criteria a record must meet to enter the approval process.
Approval Steps	Define the criteria a record must meet to enter the approval step.
Assignment Rules for Leads and Cases	Define the criteria the lead or case must meet for it to be assigned.
Auto-Response Rules for Leads and Cases	Define the criteria a lead or case must meet to trigger an auto-response rule.
Case Escalation Rules	Specify criteria a case must meet for it to be escalated.
Custom Buttons and Links	Define the content for custom links and buttons.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Reports and Approvals are not available in **Database.com**

Use Formulas for:	To:
Custom Fields	Create custom formula fields that automatically calculate a value based on other values, merge fields, or expressions. Users can view formula fields on record detail pages but can't see the underlying algorithm or edit the value of a formula field. Note: Custom formula fields are not available in Connect Offline, Web-to-Lead forms, or Web-to-Case forms.
Custom Summary Formulas in Reports	Automatically calculate more totals based on existing report summaries using the values, merge fields, or expressions you specify. Users can't change these totals.
Data Validations	Verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can include a formula such as `CloseDate >= TODAY()`.
Default Field Values	Apply a value to a custom field when a user creates a record. Use formulas to define a default value such as `TODAY() + 7`. Users can change a default value. Default field values can be based on a formula using values, merge fields, or expressions you specify.
Escalation Rules	Define the criteria that a case must meet to be escalated.
Formula Fields	Automatically calculate the value of a custom field using the values, merge fields, or expressions you specify. Users can't change the value of a formula field.
Reports	Create custom summary formulas in your reports to calculate more totals based on the existing summaries in that report.
S-Controls	Define the content for s-controls.
Validation Rules	Prevent users from entering an invalid value in a standard or custom field. Validation rules can be based on formulas and display an error message to users when the value they enter is not valid.
Workflow Field Updates	Automatically change the value of a field to a value you specify. The formula can include other values, merge fields, or expressions. You can set field updates to occur as a result of a workflow rule or an approval process.
Workflow Rules	Define the criteria a record must meet to trigger a workflow rule.
Visualforce Pages	Define the content for Visualforce pages.

Common Formula Processes

	When are they executed?	Read only?	Can include functions?	Can specify null handling?	Can include references to parent merge fields?
Default Field Values	Record creation	No	Yes	No	No
Formula Fields	Record display	Yes	Yes	Yes	Yes
Validation Rules	Record save	Not applicable	Yes	No	Yes

	When are they executed?	Read only?	Can include functions?	Can specify null handling?	Can include references to parent merge fields?
Workflow Rules	Record save	Not applicable	Yes	No	Yes
Approval Processes	Record submitted for approval	Not applicable	Yes	No	Yes
Field Updates	Workflow or approval process	Not applicable	Yes	No	Yes
Custom Summary Formulas for Reports	Report display	Yes	Yes, a limited subset of functions	Yes	No

Formula Data Types

The data type of a formula determines the type of data you expect returned from your formula.

Data Type	Description
Checkbox	Returns a true or false value. The field appears as a checkbox in record detail pages and reports. Use `True` for checked values and `False` for unchecked values.
Currency	Returns a number in currency format of up to 18 digits with a currency sign. Note: Salesforce uses the round-half-to-even tie-breaking rule for currency fields. For example, 23.5 becomes 24, 22.5 becomes 22, –22.5 becomes –22, and –23.5 becomes –24.
Date	Returns data that represents a day on the calendar. The current date can be acquired by calling the built-in function `TODAY()` in a formula. This data type is not available for custom summary formulas in reports.
Date/Time	Returns data that represents a moment in time. A date/time field includes the date and also the time of day including hour, minutes, and seconds. You can insert the current date and time in a formula using the `NOW()` function. This data type is not available for custom summary formulas in reports.
Number	Returns a positive or negative integer or decimal of up to 18 digits. Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and –12.345 becomes –12.34.
Percent	Returns a number in percent format of up to 18 digits followed by a percent sign. Percent data is stored as a decimal divided by 100, which means that 90% is equal to 0.90.
Text	Returns a string of up to 3900 characters. To display text in addition to the formula output, insert that text in quotes. Use the text data type for text, text area, URL, phone, email, address, and auto-number fields. This data type isn't available for custom summary formulas in reports.

SEE ALSO:

Build a Formula Field

Elements of a Formula

A formula can contain references to the values of fields, operators, functions, literal values, or other formulas.

Use any or all of these elements to build a formula.

Element Name	Description
Literal Value	A text string or number you enter that is not calculated or changed. For example, if you have a value that's always multiplied by 2% of an amount, your formula would contain the literal value of 2% of that amount: `ROUND((Amount*0.02), 2)` This example contains every possible part of a formula: • A function called ROUND used to return a number rounded to a specified number of decimal places. • A field reference called Amount. • An operator, *, that tells the formula builder to multiply the contents of the `Amount` field by the literal value, 0.02. • A literal number, 0.02. Use the decimal value for all percents. To include actual text in your formula, enclose it in quotes. • The last number 2 in this formula is the input required for the ROUND function that determines the number of decimal places to return.
Field Reference	Reference the value of another custom or standard field using a merge field. The syntax for a merge field is `field_name` for a standard field or `field_name__c` for a custom field. The syntax for a merge field on a related object is `object_name__r.field_name`. Use the **Insert Field** button or the drop-down list to insert a merge field in your formula where necessary.
Function	A system-defined formula that can require input from you and returns a value or values. For example, **TODAY()** does not require input but returns the current date. The **TEXT(value)** function requires your percent, number, or currency input and returns text.
Operator	A symbol that specifies the type of calculation to perform or the order in which to do it. For example, the + symbol specifies two values should be added. The open and close parentheses specify which expressions you want evaluated first.

Comment	An annotation within a formula that begins with a forward slash followed by an asterisk (`/*`). and concludes with an asterisk followed by a forward slash (`*/`). For example,
	`/*This is a formula comment*/`
	Comments are ignored when processing a formula.
	Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition. For example:
	`AND(`
	`/*competitor field is required, check to see if field is empty */`
	`LEN(Competitor__c) = 0,`
	`/* rule only enforced for ABCD record types */`
	`RecordType.Name = "ABCD Value",`
	`/* checking for any closed status, allows for additional closed picklist values in the future */`
	`CONTAINS(TEXT(StageName), "Closed")`
	`)`

Element Name	Description
	You can also use comments to *comment out* sections of your formula when debugging and checking the syntax to locate errors in the formula. Note: • Nesting comments causes a syntax error. For example, you cannot save a formula that has the following: `/* /* comment */ */` • Commenting out a whole formula causes a syntax error. • Comments count against the character and byte size limits in formulas.

SEE ALSO:

Build a Formula Field

Quick Reference Guide: Formula Fields

Formula Operators and Functions

Use these operators and functions when building formulas. All functions are available everywhere that you can include a formula—such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Note: Extraneous spaces in the samples below are ignored.

- Math Operators
- Logical Operators
- Text Operators
- Date and Time Functions
- Informational Functions
- Logical Functions
- Math Functions
- Text Functions
- Summary Functions
- Advanced Functions
- Encoding Functions

Math Operators

Operator	Description
+ (Add)	Calculates the sum of two values.
- (Subtract)	Calculates the difference of two values.
*** (Multiply)**	Multiplies its values.

Operator	Description
/ (Divide)	Divides its values.
^ (Exponentiation)	Raises a number to a power of a specified number.
() (Open Parenthesis and Close Parenthesis)	Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.

Logical Operators

Operator	Description
= and == (Equal)	Evaluates if two values are equivalent. The = and == operator are interchangeable.
<> and != (Not Equal)	Evaluates if two values are not equivalent.
< (Less Than)	Evaluates if a value is less than the value that follows this symbol.
> (Greater Than)	Evaluates if a value is greater than the value that follows this symbol.
<= (Less Than or Equal)	Evaluates if a value is less than or equal to the value that follows this symbol.
>= (Greater Than or Equal)	Evaluates if a value is greater than or equal to the value that follows this symbol.
&& (AND)	Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.
|| (OR)	Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

Text Operators

Operator	Description
& (Concatenate)	Connects two or more strings.

Date and Time Functions

Function	Description
DATE	Returns a date value from year, month, and day values you enter. Salesforce displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.
DATEVALUE	Returns a date value for a date/time or text expression.
DATETIMEVALUE	Returns a year, month, day and GMT time value.
DAY	Returns a day of the month in the form of a number between 1 and 31.
MONTH	Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.

Function	Description
NOW	Returns a date/time representing the current moment.
TODAY	Returns the current date as a date data type.
YEAR	Returns the four-digit year in number format of a given date.

Informational Functions

Function	Description
BLANKVALUE	Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.
ISBLANK	Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.
ISNULL	Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE. Important: Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce will continue to support ISNULL, so you do not need to change any existing formulas.
NULLVALUE	Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression. Important: Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Salesforce will continue to support NULLVALUE, so you do not need to change existing formulas.
PRIORVALUE	Returns the previous value of a field.

Logical Functions

Function	Description
AND	Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false.
CASE	Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the `else_result`.
IF	Determines if expressions are true or false. Returns a given value if true and another value if false.
ISCHANGED	Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.
ISNEW	Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.
ISNUMBER	Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.

Function	Description
NOT	Returns FALSE for TRUE and TRUE for FALSE.
OR	Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false.

Math Functions

Function	Description
ABS	Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.
CEILING	Rounds a number up to the nearest integer.
DISTANCE	Calculates the distance between two locations in miles or kilometers.
EXP	Returns a value for e raised to the power of a number you specify.
FLOOR	Returns a number rounded down to the nearest integer.
GEOLOCATION	Returns a geolocation based on the provided latitude and longitude. Must be used with the DISTANCE function.
LN	Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.
LOG	Returns the base 10 logarithm of a number.
MAX	Returns the highest number from a list of numbers.
MIN	Returns the lowest number from a list of numbers.
MOD	Returns a remainder after a number is divided by a specified divisor.
ROUND	Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.
SQRT	Returns the positive square root of a given number.

Text Functions

Function	Description
BEGINS	Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not.
BR	Inserts a line break in a string of text.
CASESAFEID	Converts a 15-character ID to a case-insensitive 18-character ID.
CONTAINS	Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.

Function	Description
FIND	Returns the position of a string within a string of text represented as a number.
GETSESSIONID	Returns the user's session ID.
HYPERLINK	Creates a link to a URL specified that is linkable from the text specified.
IMAGE	Inserts an image with alternate text and height/width specifications.
INCLUDES	Determines if any value selected in a multi-select picklist field equals a text literal you specify.
ISPICKVAL	Determines if the value of a picklist field is equal to a text literal you specify.
LEFT	Returns the specified number of characters from the beginning of a text string.
LEN	Returns the number of characters in a specified text string.
LOWER	Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
LPAD	Inserts characters you specify to the left-side of a text string.
MID	Returns the specified number of characters from the middle of a text string given the starting position.
RIGHT	Returns the specified number of characters from the end of a text string.
RPAD	Inserts characters that you specify to the right-side of a text string.
SUBSTITUTE	Substitutes new text for old text in a text string.
TEXT	Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, validation rules, formula fields, field updates, and custom buttons and links.
TRIM	Removes the spaces and tabs from the beginning and end of a text string.
UPPER	Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
VALUE	Converts a text string to a number.

Summary Functions

The following functions are available with summary, matrix, and joined reports.

Function	Description
PARENTGROUPVAL	This function returns the value of a specified parent grouping. A "parent" grouping is any level above the one containing the formula. You can only use this function in custom summary formulas for reports.
PREVGROUPVAL	This function returns the value of a specified previous grouping. A "previous" grouping is one that comes before the current grouping in the report. Choose the grouping level and increment. The increment is the number of columns or rows before the current summary. The default is 1; the maximum is 12. You can only use this function in custom summary formulas for reports.

Advanced Functions

Function	Description
GETRECORDIDS	Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.
INCLUDE	Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.
LINKTO	Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce page.
REGEX	Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, it returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.
REQUIRESCRIPT	Returns a script tag with source for a URL you specify. Use this function when referencing the Force.com AJAX Toolkit or other JavaScript toolkits.
URLFOR	Returns a relative URL for an action, s-control, Visualforce page, or a file in a static resource archive in a Visualforce page.
VLOOKUP	Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.

Encoding Functions

Function	Description
HTMLENCODE	Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (>), with HTML entity equivalents, such as `>`.
JSENCODE	Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (').
JSINHTMLENCODE	Encodes text and merge field values for use in JavaScript inside HTML tags by replacing characters that are reserved in HTML with HTML entity equivalents and inserting escape characters before unsafe JavaScript characters. `JSINHTMLENCODE(`***`someValue`***`)` is a convenience function that is equivalent to `JSENCODE(HTMLENCODE(`***`someValue`***`))`. That is, `JSINHTMLENCODE` first encodes *`someValue`* with `HTMLENCODE`, and then encodes the result with `JSENCODE`.
URLENCODE	Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in *RFC 3986, Uniform Resource Identifier (URI): Generic Syntax*. For example, blank spaces are replaced with `%20`, and exclamation points are replaced with `%21`.

IN THIS SECTION:

Formula Operators and Functions A–H

Formula Operators and Functions I–Z

SEE ALSO:

Examples of Advanced Formula Fields

Formula Operators and Functions A–H

Use the following operators and functions when building formulas. Click on the name of the operator or function below to view more details. All functions are available everywhere that you can include a formula such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Note: Extraneous spaces in the samples below are ignored.

+ (Add)

Description:	Calculates the sum of two values.
Use:	`value1 + value2` and replace each *`value`* with merge fields, expressions, or other numeric values.
Formula Field Example:	`Amount + Maint_Amount__c + Services_Amount__c` This formula calculates the sum of the product `Amount`, maintenance amount, and services fees. Note that `Maint amount` and `Service Fees` are custom currency fields.
Report Example:	`EMAIL_OPT_OUT:SUM + DO_NOT_CALL:SUM` calculates all `Email Opt Out` fields plus all `Do Not Call` fields on the leads in your report. This formula is a number data type that returns a positive integer.
Validation Rule Example:	You may have a custom object that allows users to track the total number of hours worked in a week. Use the following example to ensure that users cannot save a time card record with more than 40 hours in a work week. `Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40` Use a formula like this one in a validation rule to display the following error message when the total number of hours entered for each work day is greater than 40: "Your total hours cannot exceed 40." This example requires five custom fields on your custom object, one for each day of work.

- (Subtract)

Description:	Calculates the difference of two values.
Use:	`value1 - value2` and replace each *`value`* with merge fields, expressions, or other numeric values.

Example:	`Amount - Discount_Amount__c` This formula calculates the difference of the product `Amount` less the `Discount Amount`. Note that `Discount Amount` is a custom currency field.
Report Example:	`AMOUNT:SUM - Product.Discount_Amount__c:SUM` calculates the difference of all `Amount` fields and all `Discounted Amount` custom fields on the products in your report. This formula is a currency data type that returns a currency sign and decimal places.

* (Multiply)

Description:	Multiplies its values.
Use:	`value1 * value2` and replace each *`value`* with merge fields, expressions, or other numeric values.
Example:	`Consulting_Days__c * 1200` This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of $1200 per day. Note that `Consulting Days` is a custom field.
Report Example:	`RowCount * AGE:AVG` calculates the record count times the average age value of your report. This formula is a number data type that returns a positive or negative integer or decimal.

/ (Divide)

Description:	Divides its values.
Use:	`value1 / value2` and replace each *`value`* with merge fields, expressions, or other numeric values.

Example:	`AnnualRevenue/ NumberOfEmployees`
	This formula calculates the revenue amount per employee using a currency field.
	`IF(NumberOfOpportunities > 0, NumberOfWonOpportunities / NumberOfOpportunities, null)`
	This formula calculates the win rate of opportunities on a campaign.

Report Example:	**% Won Opportunities** `WON:SUM / RowCount` calculates the percent of `Won` opportunities using a record count representing the number of all opportunities in your report. This formula is a number data type that returns a positive or negative integer. **% Difference between Cost and Sales Price** `(TOTAL_PRICE:SUM - QUANTITY:SUM * Product2.Cost__c:SUM) / (QUANTITY:SUM * Product2.Cost__c:SUM)` calculates the average percent difference between what a product costs and its selling price on a product-by-product level. Note that `Product2.Cost__c:SUM` is a custom currency field named `Cost` on products, which includes

	the cost of each product. This formula is a percent data type that returns a positive or negative integer. For best results, use this on a summary Opportunities with Products report that is summarized by `Product Name` and includes summary totals for `Quantity`, `Total Price`, and `Cost`.

^ (Exponentiation)

Description:	Raises a number to a power of a specified number.
Use:	*`number^integer`* and replace *`number`* with a merge field, expression, or another numeric value; replace *`integer`* with a merge field that contains an integer, expression, or any integer.
Example:	`NumberOfEmployees^4` calculates the number of employees to the 4th power.
Report Example:	`ACTIVE:SUM ^ 2` calculates the number of active Salesforce users to the 2nd power for administration. This formula is a number data type that returns a positive integer.
Tips:	Avoid replacing *`integer`* with a negative number.

() (Open Parenthesis and Close Parenthesis)

Description:	Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.
Use:	*`(expression1) expression2...`* and replace each *`expression`* with merge fields, expressions, or other numeric values.
Example:	`(Unit_Value__c - Old_Value__c) / New_Value__c` calculates the difference between the old value and new valuedivided by the new value.
Report Example:	`(DURATIONHOURS:SUM * RowCount) / 24` calculates the duration of all event times the record count per 24 hours. This formula is a percent data type that returns a positive or negative integer or decimal, representing what percent of a day is spent on events.

= and == (Equal)

Important: Don't use this function for a null comparison, such as `MyDateTime__c == null`. Use ISBLANK instead.

Description:	Evaluates if two values are equivalent. The = and == operator are interchangeable.
Use:	*`expression1=expression2`* or *`expression1 == expression2`*, and replace each *`expression`* with merge fields, expressions, or other numeric values.
Example:	**Due Date** `Due Date = CreatedDate + 5` returns true if the due date is equal to five days following a record's created date. **Commission Amount** `IF(Probability =1, ROUND(Amount*0.02, 2), 0)`

This formula calculates the 2% commission amount of an opportunity that has a probability of 100%. All other opportunities will have a commission value of 0.

Possible results:

- An opportunity with a `Probability` of 90% will have a commission of 0.
- An opportunity with a `Probability` of 100% and an `Amount` of $100,000 will have a commission of $2,000.

<> and != (Not Equal)

Important: Don't use this function for a null comparison, such as `MyDateTime__c != null`. Use ISBLANK instead.

Description:	Evaluates if two values are not equivalent.
Use:	*`expression1 <> expression2`* or *`expression1 != expression2`*, and replace each `expression` with merge fields, expressions, or other numeric values.
Example:	`IF(Maint_Amount__c + Services_Amount__c<> Amount, "DISCOUNTED", "FULL PRICE")` This formula displays "DISCOUNTED" on product if its maintenance amount and services amount do not equal the product amount. Otherwise, displays "FULL PRICE." Note that this example uses two custom currency fields for `Maint Amount` and `Services Amount`.

< (Less Than)

Description:	Evaluates if a value is less than the value that follows this symbol.
Use:	*`value1 < value2`* and replace each `value` with merge fields, expressions, or other numeric values.
Example:	`IF(AnnualRevenue < 1000000, 1, 2)` assigns the value "1" with revenues less than one million and the value "2" to revenues greater than one million.

> (Greater Than)

Description:	Evaluates if a value is greater than the value that follows this symbol.
Use:	*`value1 > value2`* and replace each *`value`* with merge fields, expressions, or other numeric values.
Example:	`IF(commission__c > 1000000, "High Net Worth", "General")` assigns the "High Net Worth" value to a commission greater than one million. Note, this is a text formula field that uses a commission custom field.

<= (Less Than or Equal)

Description:	Evaluates if a value is less than or equal to the value that follows this symbol.
Use:	*value1* `<=` *value2* and replace each `value` with merge fields, expressions, or other numeric values.
Example:	`IF(AnnualRevenue <= 1000000, 1, 2)` assigns the value "1" with revenues less than or equal to one million and the value "2" with revenues greater than one million.

>= (Greater Than or Equal)

Description:	Evaluates if a value is greater than or equal to the value that follows this symbol.
Use:	*value1* `>=` *value2* and replace each `value` with merge fields, expressions, or other numeric values.
Example:	`IF(Commission__c >= 1000000, "YES", "NO")` assigns the "YES" value with a commission greater than or equal to one million. Note, this is a text formula field that uses a custom currency field called `Commission`.

&& (AND)

Description:	Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.
Use:	(***logical1***) `&&` (***logical2***) and replace *logical1* and *logical2* with the values or expressions that you want evaluated.
Example:	`IF((Price<100 && Quantity<5),"Small", null)` This formula displays "Small" if the price is less than 100 and quantity is less than five. Otherwise, this field is blank.

|| (OR)

Description:	Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.
Use:	(***logical1***) `||` (***logical2***) and replace any number of logical references with the values or expressions you want evaluated.
Example:	`IF((ISPICKVAL(Priority, "High")) || (ISPICKVAL(Status , "New")), ROUND(NOW()-CreatedDate, 0), null)` This formula returns the number of days a case has been open if the `Status` is new or the `Priority` is high. If the case was opened today, this field displays a zero.
Validation Rule Example:	`(Discount_Rate__c < 0) || (Discount_Rate__c > 0.40)`

	This validation rule formula displays the following error message when the `Discount Rate` custom field is not between 0 and 40%: "Discount Rate cannot exceed 40%."

& (Concatenate)

Description:	Connects two or more strings.
Use:	*`string1&string2`* and replace each *`string`* with merge fields, expressions, or other values.
Example:	`"Expense-" & Trip_Name__c & "-" & ExpenseNum__c` This formula displays the text "Expense-" followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

ABS

Description:	Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.
Use:	*`ABS(number)`* and replace *`number`* with a merge field, expression, or other numeric value that has the sign you want removed.
Example:	`ABS(ExpectedRevenue)` calculates the positive value of the `Expected Revenue` amount regardless of whether it is positive or negative.

AND

Description:	Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false. Use this function as an alternative to the operator && (AND).
Use:	`AND(logical1,logical2,...)` and replace *`logical1,logical2,...`* with the values that you want evaluated.
Formula Field Example:	`IF(AND(Price<1,Quantity<1),"Small", null)` This formula displays "Small" if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

BEGINS

Description:	Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not.
Use:	`BEGINS(`***`text`***`, compare_text)` and replace *`text, compare_text`* with the characters or fields you want to compare.

Example:	`IF(BEGINS (Product_type__c, "ICU"), "Medical", "Technical")` This example returns the text "Medical" if the text in any `Product Type` custom text field begins with "ICU." For all other products, it displays "Technical."
Tips:	• This function is case sensitive so be sure your *`compare_text`* value has the correct capitalization. • When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset begins with "3," all assets that have a blank serial number are considered valid.

BLANKVALUE

Description:	Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.
Use:	`BLANKVALUE(`***`expression, substitute_expression`***`)` and replace *`expression`* with the expression you want evaluated; replace *`substitute_expression`* with the value you want to replace any blank values.

Example:	Example 1
	`BLANKVALUE(Department, "Undesignated")`
	This formula returns the value of the `Department` field if the `Department` field contains a value. If the `Department` field is empty, this formula returns the word `Undesignated`.
	Example 2
	`(BLANKVALUE(Payment_Due_Date__c, StartDate +5)`
	This formula returns the date five days after the contract start date whenever `Payment Due Date` is blank. `Payment Due Date` is a custom date field.

Tips:	• Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Salesforce will continue to support NULLVALUE, so you do not need to change existing formulas. • A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty. • Use the BLANKVALUE function to return a specified string if the field does not have a value; use the ISBLANK function if you only want to check if the field has a value. • If you use this function with a numeric field, the function only returns the specified string if the field does not have a value and is not configured to treat blank fields as zeroes.

BR

Description:	Inserts a line break in a string of text.
Use:	`BR()`

Example:	`CASE(ShippingCountry,`
	`"USA",`
	`  ShippingStreet & BR() &`
	`  ShippingCity & ",`
	`  " & ShippingState & " " &`
	`  ShippingPostalCode & BR()`
	`  & ShippingCountry,`
	`"France",`
	`  ShippingStreet & BR() &`
	`  ShippingPostalCode & " " &`
	`  ShippingCity & BR() &`
	`  ShippingCountry, "etc")`
	
	This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country.

Tips:	• Do not remove the parentheses after the function name. • Keep the parentheses empty. They do not need to contain a value. • Remember to surround the BR() with concatenation operators: &. • Avoid using this function in mail merge templates. • This function is not available in custom buttons and links, s-controls, or reports.

CASE

Description:	Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the `else_result`.
Use:	`CASE(`***`expression`***`,`***`value1`***`,` ***`result1`***`,` ***`value2`***`,` ***`result2`***`,...,` ***`else_result`***`)` and replace *`expression`* with the field or value you want compared to each specified value. Replace each value and result with the value that must be equivalent to return the result entry. Replace *`else_result`* with the value you want returned when the expression does not equal any values.

Formula Field Example:	**Days Open for Cases**
	
	Use this example of a custom formula field called `Days Open` to display different text depending on the number of days a case has been open:
	
	`CASE(Days_Open__c, 3,`
	`  "Reassign", 2, "Assign Task", "Maintain")`
	
	The following text is displayed:
	• "Reassign" for any case open three days.
	• "Assign Task" for any case open two days.
	• "Maintain" for all other cases.
	
	Last Activity Month

This formula field displays the month of the last activity or "None" if there are no activities.

```
CASE(MONTH(LastActivityDate),
1, "January",
2, "February",
3, "March",
4, "April",
5, "May",
6, "June",
7, "July",
8, "August",
9, "September",
10, "October",
11, "November",
12, "December",
"None")
```

Default Value Example:

Discount Rate

Use the following default value formula to insert a different discount rate on an opportunity based on the department of the person creating the opportunity:

```
CASE(User.Department, "IT", 0.25, "Field", 0.15, 0)
```

In this example, the formula inserts a discount rate of 25% on any opportunity created by a user in the "IT" department or 15% on any opportunity created by someone in the "Field" department. A zero is applied if the creator does not belong to either of these departments. This is a custom percent field on opportunities that uses the standard user field `Department`.

Product Language

You may want to associate a product with its language so that your users know the type of documentation or adapter to include. Use the following default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is "Japanese" if the user's country is "Japan" and "English" if the user's country is "US." If neither is true, the default value "unknown" is inserted into the `Product Language` field.

```
CASE($User.Country , "Japan", "Japanese", "US",
"English","unknown")
```

Tips:

- Be sure your *value1*, *value2*... expressions are the same data type.
- Be sure your *result1*, *result2*... expressions are the same data type.
- CASE functions cannot contain functions that return true or false. Instead, make true or false expressions return numbers such as:

  ```
  CASE(1, IF(ISPICKVAL (Term__c, "12"), 1, 0),
   12 * Monthly_Commit__c,
   IF(ISPICKVAL(Term__c, "24"), 1, 0),
   24 * Monthly_Commit__c, 0)
  ```

  In this formula, `Term` is a picklist field that is multiplied by the Monthly Commit whenever it contains the value 1 for true.

- The *else_result* value is required.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one should be returned. For example, `CASE(Field__c,"Partner", "P", "Customer", "C", LEFT(Field__c, -5))` returns an error even if the value of the field is "Partner" or "Customer" because the last statement is illogical.
- If the field in your CASE function is blank, it returns your *else_result* value. For example, this formula: `CASE(Days_Open__c, 3, "Reassign", 2, "Assign Task", "Maintain")` displays "Maintain" if the `Days Open` field is blank, 0, or any value other than 2 or 3.
- Use CASE functions to determine if a picklist value is equal to a particular value. For example the formula `CASE(Term__c, "12", 12 * Monthly_Commit__c, "24", 24 * Monthly_Commit__c, 0)` multiplies the `Monthly Commit` amount by 12 whenever the `Term` is 12 or multiplies the `Monthly Commit` amount by 24 whenever the `Term` is 24. Otherwise, the result is zero.

CASESAFEID

Description:	Converts a 15-character ID to a case-insensitive 18-character ID.
Use:	`CASESAFEID(`***id***`)` and replace *id* with the object's ID.
Example:	`CASESAFEID (Id)` This formula replaces the 15-character ID with the 18-character, case-insensitive ID.
Tips:	• Convert to 18-character IDs for better compatibility with Excel. • The CASESAFEID function is available everywhere that you can define a formula except reports and s-controls.

CEILING

Description:	Rounds a number up to the nearest integer.
Use:	`CEILING(`***number***`)` and replace *number* with the field or expression you want rounded.
Example:	**Rounding Up (literal value)** `CEILING(2.5)` This formula returns 3, which is 2.5 rounded up to the nearest number. **Earthquake Magnitude** `CEILING(Magnitude__c)` returns the value of a formula number field that calculates the magnitude of an earthquake up to the nearest integer.

CONTAINS

Description:	Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.
Use:	`CONTAINS(text, compare_text)` and replace *`text`* with the text that contains the value of *`compare_text`*.
Example:	`IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")` This formula checks the content of a custom text field named `Product_Type` and returns "Parts" for any product with the word "part" in it. Otherwise, it returns "Service." Note that the values are case sensitive, so if a `Product_Type` field contains the text "Part" or "PART," this formula returns "Services."
Tips:	• This function is case sensitive so be sure your *`compare_text`* value has the correct capitalization. • When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset contains "A," all assets that have a blank serial number are considered valid. • The CONTAINS function does not support multi-select picklists. Use INCLUDES to see if a multi-select picklist has a specific value.

DATE

Description:	Returns a date value from year, month, and day values you enter. Salesforce displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.
Use:	`DATE(year,month,day)` and replace *`year`* with a four-digit year, *`month`* with a two-digit month, and *`day`* with a two-digit day.
Example:	`DATE(2005, 01, 02)` creates a date field of January 2, 2005.

DATEVALUE

Description:	Returns a date value for a date/time or text expression.
Use:	`DATEVALUE(expression)` and replace *`expression`* with a date/time or text value, merge field, or expression.
Example:	**Closed Date** `DATEVALUE(ClosedDate)` displays a date field based on the value of the `Date/Time Closed` field. **Date Value** `DATEVALUE("2005-11-15")` returns November 15, 2005 as a date value.

Tips:	• If the field referenced in the function is not a valid text or date/time field, the formula field displays #ERROR! • When entering a date, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day. • If the `expression` does not match valid date ranges, such as the MM is not between 01 and 12, the formula field displays #ERROR! • Dates and times are always calculated using the user's time zone, except in list views, reports, and related lists. These items calculate dates and times using Coordinated Universal Time.

DATETIMEVALUE

Description:	Returns a year, month, day and GMT time value.
Use:	`DATETIMEVALUE(`***`expression`***`)` and replace *`expression`* with a date/time or text value, merge field, or expression.
Example:	**Closed Date** `DATETIMEVALUE(ClosedDate)` displays a date field based on the value of the `Date/Time Closed` field. **Date Value** `DATETIMEVALUE("2005-11-15 17:00:00")` returns November 15, 2005 5:00 PM GMT as a date and time value.
Tips:	• DATETIMEVALUE is always calculated using GMT time zone and can't be changed. • When entering a specific date, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day. • If the *`expression`* does not match valid date ranges, such as the MM is not between 01 and 12, the formula field displays #ERROR!

DAY

Description:	Returns a day of the month in the form of a number between 1 and 31.
Use:	`DAY(`***`date`***`)` and replace *`date`* with a date field or value such as *`TODAY()`*.
Example:	`DAY(Code_Freeze__c)` returns the day in your custom code freeze date. Note this does not work on date/time fields.

DISTANCE

Description:	Calculates the distance between two locations in miles or kilometers.

Use:

`DISTANCE(mylocation1, mylocation2, 'unit')` and replace *mylocation1* and *mylocation2* with two location fields, or a location field and a value returned by the GEOLOCATION function. Replace *unit* with mi (miles) or km (kilometers).

Examples:

Distance between two geolocation fields

```
DISTANCE(warehouse_location__c, store_location__c, 'mi')
```

This formula returns the distance, in miles, between the warehouse and the store. In this example, `warehouse_location__c` and `store_location__c` are the names of two custom geolocation fields.

Distance between an address field and a geolocation field

```
DISTANCE(BillingAddress, store_location__c, 'mi')
```

This formula returns the distance, in miles, between an account's billing address and a store. In this example, `BillingAddress` is the standard billing address field on an Account object, and `store_location__c` is the name of a custom geolocation field.

Distance between a custom geolocation field and fixed coordinates

```
DISTANCE(warehouse_location__c, GEOLOCATION(37.775,-122.418),
'km')
```

This formula returns the distance, in kilometers, between the warehouse and the known latitude and longitude 37.775°, -122.418° (San Francisco).

Distances with conditions

```
IF(DISTANCE(warehouse_location__c, ShippingAddress, 'mi')<10,
"Near", "Far")
```

This formula updates a text formula field to `Near` if the distance between the warehouse and the account shipping address compound field is less than 10 miles. Otherwise, it updates the text field to `Far`.

Tip: Although DISTANCE can be calculated in miles or kilometers, the unit is not returned in the calculation. If possible, include the unit of measure in the name of your distance formula field, so users know whether the distance is in miles or kilometers.

Tips:

- The DISTANCE function returns a number data type. Distance is always calculated in decimals, even if you're displaying the geolocation notation in degrees, minutes, and seconds in the user interface. Specify the number of decimal places to show when you create a custom field.
- The DISTANCE function isn't available in reports, but it can be used in list views. To use DISTANCE in your reports, set up a formula field, and then reference the field in your reports.
- DISTANCE is the only formula function that can use GEOLOCATION parameters.
- There are limitations on DISTANCE accuracy and equality calculations.
  - DISTANCE supports only the logical operators > and <, returning values within (<) or beyond (>) a specified radius.
  - Distance is calculated as a straight line—as the crow flies—regardless of geography and topography between the two points.

  For additional details, see "How SOQL Calculates and Compares Distances" in the *Force.com SOQL and SOSL Reference*.

EXP

Description:	Returns a value for e raised to the power of a number you specify.
Use:	`EXP(number)` and replace *`number`* with a number field or value such as 5.

Example:	**Exponent of a Literal Value**
	`EXP(3)`
	This formula returns the value of e to the third power.
	Compound Interest
	`Principal__c * EXP(Rate__c * Years__c)`
	This formula calculates the compound interest based on a custom currency field for principal, custom percent field for rate, and custom number field for years.

FIND

Description:	Returns the position of a string within a string of text represented as a number.
Use:	`FIND(search_text, text[, start_num])` and replace *`search_text`* with the string you want to find, replace *`text`* with the field or expression you want to search, and replace *`start_num`* with the number of the character from which to start searching from left to right.
Example:	**Street Address** `FIND(" ", Street)` returns the character position of the first space in the `Street` field. You can use this number to find out the length of the street address as a means of separating a street address from street name in an address field. **Deriving Website Addresses** `SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.")` finds the location of the @ sign in a person's email address to determine the length of text to replace with a "www." as a means of deriving their website address.
Tips:	• Be sure to remove the brackets, [and], from your formula before validating it. • If the field referenced in your *`text`* parameter is blank, the formula field displays 0. • Your *`search_text`* parameter is case sensitive and cannot contain any wildcard characters. • If your search does not return any results, a 0 displays in the field. • The *`start_num`* parameter is optional. If you do not enter a *`start_num`* value, the formula uses the value one, or the first character in the string. • If your *`start_num`* is not greater than zero, a 0 displays in the field. • If your *`start_num`* is greater than the length of the text, a 0 displays in the field. • When entering your *`start_num`* parameter, remember that some fields like the `Website` field are unique because a "http://" is automatically appended to the beginning of the text you enter. • Note that the first character in a string is designated as one rather than zero.

FLOOR

Description:	Returns a number rounded down to the nearest integer.
Use:	`FLOOR(number)` and replace *number* with a number field or value such as 5.245.
Example:	**Commission Amounts** `FLOOR(commission__c)` rounds commission down to the nearest integer.

GEOLOCATION

Description:	Returns a geolocation based on the provided latitude and longitude. Must be used with the DISTANCE function.
Use:	`GEOLOCATION(latitude, longitude)` and replace *latitude* and *longitude* with the corresponding geolocation, numerical code values.
Examples:	**Distance between a custom geolocation field and fixed coordinates** `DISTANCE(warehouse_location__c, GEOLOCATION(37.775,-122.418), 'km')` This formula returns the distance, in kilometers, between the warehouse and the known latitude and longitude 37.775°, -122.418° (San Francisco).
Tips:	• The GEOLOCATION function returns a location data type that can be used only by, and must be used with, the DISTANCE function. The GEOLOCATION function doesn't work on its own.

GETRECORDIDS

Description:	Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.
Use:	`{!GETRECORDIDS(object_type)}` and replace *object_type* with a reference to the custom or standard object for the records you want to retrieve.
Custom Button Example:	`{!REQUIRESCRIPT ("/soap/ajax/13.0/connection.js")} var records = {!GETRECORDIDS($ObjectType.Sample)}; var newRecords = []; if (records[0] == null) { alert("Please select at least one row") } else { for (var n=0; n<records.length; n++) { var c = new sforce.SObject("Case"); c.id = records[n]; c.Status = "New"; newRecords.push(c); } result = sforce.connection.update(newRecords); window.location.reload(); }` In this example, all selected case records are updated with a `Status` of "New." To set this up in your organization, create a custom list button for cases with the following attributes: • `Display Type` is "List Button" • `Behavior` is "Execute JavaScript"

- `Content Source` is "OnClick JavaScript"

Paste the sample code above into the content of your custom button. Finally, add the list button to the a page layout that contains the Cases related list, such as accounts or opportunities. Users can select any number of cases in the related list and click the list button to change the status of those cases at once. Notice the check for `records[0] == null`, which displays a message to users when they do not select at least one record in the list.

Tips:

- Use global variables to access special merge fields for s-controls, custom buttons, and links.
- Activities are special types of objects. Use {!GETRECORDIDS($ObjectType.Task)} when creating a task list button. Use {!GETRECORDIDS($ObjectType.Event)} when creating an event list button.
- This function is only available in custom buttons, links, and s-controls.

GETSESSIONID

Description: Returns the user's session ID.

Use: `GETSESSIONID()`

Example:

```
HYPERLINK
("https://www.myintegration.com?sId="&
GETSESSIONID() & "?&rowID="&Name & "action=CreateTask","Create
a Meeting Request")
```

creates a link to an application outside of Salesforce, passing the parameters so that it can connect to Salesforce via the API and create the necessary event.

Tips:

Important: `$Api.Session_ID` and `GETSESSIONID()` return the same value, an identifier for the current session in the current context. This context varies depending on where the global variable or function is evaluated. For example, if you use either in a custom formula field, and that field is displayed on a standard page layout in Salesforce Classic, the referenced session will be a basic Salesforce session. That same field (or the underlying variable or formula result), when used in a Visualforce page, references a Visualforce session instead.

Session contexts are based on the domain of the request. That is, the session context changes whenever you cross a hostname boundary, such as from `.salesforce.com` to `.visual.force.com` or `.lightning.force.com`.

Session identifiers from different contexts, and the sessions themselves, are different. When you transition between contexts, the old session is replaced by the new one, and the old session is no longer valid. The session ID also changes at this time.

Normally Salesforce transparently handles session hand-off between contexts, but if you're passing the session ID around yourself, be aware that you might need to re-access `$Api.Session_ID` or `GETSESSIONID()` from the new context to ensure a valid session ID.

Note also that not all sessions are created equal. In particular, sessions obtained in a Lightning Experience context have reduced privileges, and don't have API access. You can't use these session IDs to make API calls.

HTMLENCODE

Description:	Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (>), with HTML entity equivalents, such as `>`.
Use:	`{!HTMLENCODE(text)}` and replace *`text`* with the merge field or text string that contains the reserved characters.
Example:	If the merge field `foo__c` contains `<B>Enter the user's name<b>`, `{!HTMLENCODE(foo__c)}` results in: `<B>Enter the user's name</b>`
Tips:	This function is only available in custom buttons and links, and in Visualforce.

HYPERLINK

Description:	Creates a link to a URL specified that is linkable from the text specified.
Use:	`HYPERLINK(url, friendly_name [,target])` and replace *`url`* with the Web address, replace *`friendly_name`* with the link text, and, optionally, replace *`target`* with the window or frame in which to display the content.
Example:	**Creating Events** `HYPERLINK("00U/e?retURL=%2F006x0000001T8Om&what_id=" & Id, "Create Event")` adds a link called "Create Event" that, when clicked, creates a new event that is associated with the current object. **Phone Dialer** `HYPERLINK("http://servername/call?id=" & Id & "&phone=" & Phone, Phone)` creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace `"servername"` and `"call"` with the name of your dialing tool and the command it uses to dial. The merge field, `Id`, inserts the identifier for the contact, lead, or account record. The first `Phone` merge field tells the dialing tool what number to call and the last `Phone` merge field uses the value of the `Phone` field as the linkable text the user clicks to dial.
Tips:	• Hyperlink formula fields are of type text. • Include the protocol and URL in quotes as in `HYPERLINK("http://www.cnet.com", "cnet")`. • Avoid using text functions such as LEN, LEFT, or RIGHT on HYPERLINK function results. • When linking to Salesforce pages, use a relative link, such as "00U/e?retURL=%...", for hyperlink formulas unless you want to add the formula field to a search layout. Use the complete URL, including the server name and https://, in a hyperlink formula to add it to a search layout. Note that formula fields are not available in search result layouts. • Use the `$Api` variable to reference API URLs.

- Be sure to remove the brackets, [and], from your formula before validating it.
- The *target* parameter is optional. If you do not specify a *target*, the link opens in a new browser window. Some common *target* parameters are:

  _blank
  Displays link in a new unnamed window.

  _self
  Displays link in the same frame or window as the element that refers to it.

  _parent
  Displays link in the immediate frameset parent of the current frame. This value is the same as _self if the current frame has no parent.

  _top
  Displays link in the full original window, canceling any other frames. This value is the same as _self if the current frame has no parent.

  For more information on basic HTML tags, consult an HTML reference on the Internet.
- The HYPERLINK function is available everywhere that you can define a formula except default values, field updates, s-controls, validation rules, approval processes, custom buttons and links, and workflow rules.

SEE ALSO:

Formula Operators and Functions I–Z
Formula Operators and Functions

Formula Operators and Functions I–Z

Use the following operators and functions when building formulas. Click on the name of the operator or function below to view more details. All functions are available everywhere that you can include a formula such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Note: Extraneous spaces in the samples below are ignored.

IF

Description:	Determines if expressions are true or false. Returns a given value if true and another value if false.
Use:	`IF(logical_test, value_if_true, value_if_false)` and replace *logical_test* with the expression you want evaluated; replace *value_if_true* with the value you want returned if the expression is true; replace *value_if_false* with the value you want returned if the expression is false.
Formula Field Example:	**Overdue Payments** `IF(AND(Payment_Due_Date__c < TODAY(),Payment_Status__c ="UNPAID"), "PAYMENT OVERDUE", null)`

This formula determines if the payment due date is past and the payment status is "UNPAID." If so, returns the text "PAYMENT OVERDUE" and if not, leaves the field blank. This example uses a custom date field called `Payment Due Date` and a text custom field called `Payment Status`.

Insert Tax Rate

Use this default value formula to set the tax rate of an asset based on the user's city. Create a custom percent field with the following default value:

```
IF($User.City = "Napa", 0.0750,
 IF($User.City = "Paso Robles", 0.0725,
  IF($User.City = "Sutter Creek", 0.0725,
   IF($User.City = "Los Olivos", 0.0750,
    IF($User.City = "Livermore", 0.0875, null
    )
   )
  )
 )
)
```

Custom Button Example:

```
{!
IF(Sample.BillingCountry = "US",
"http://maps.google.com/maps?q="&Sample.BillingStreet&
"+"&Sample.BillingCity&"+"&Sample.BillingState&"+
"&Sample.BillingCountry,
(IF(Sample.BillingCountry = "UK",
"http://maps.google.co.uk/maps?q="&Sample.BillingStreet
&"+"&Sample.BillingCity&"+"&Sample.BillingCountry,
"http://maps.google.com")))
}
```

This example uses the IF function to determine if an address is in the United States or United Kingdom so that it can use the appropriate type of Google map to display the address.

Tips:

- Make sure your *value_if_true* and *value_if_false* expressions are the same data type.
- When using an IF function with the $Profile.UserType variable to determine the type of Salesforce user license the logged in user has, use the following values:
  - Standard for Salesforce
  - PowerPartner for PRM User
  - CustomerSuccess for Customer Portal User
  - PowerCustomerSuccess for Customer Portal Manager

  For example, use the following formulas to determine if the logged in user has the license type in quotes:

  ```
  IF(ISPICKVAL($Profile.UserType ,"Standard"), 100, 0.1)
  ```

  ```
  IF(ISPICKVAL($Profile.UserType ,"PowerPartner"), 100, 0.1)
  ```

  ```
  IF(ISPICKVAL($Profile.UserType ,"CustomerSuccess"), 100, 0.1)
  ```

Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.

IMAGE

Description:	Inserts an image with alternate text and height/width specifications.
Use:	`IMAGE(image_url, alternate_text, height, width)` and replace `image_url` with the full path to the image; replace `alternate_text` with the string of text you want to appear when the image can't be rendered for some reason, and which can be used by screen reader software; replace `height` with the vertical size of the image in pixels; replace `width` with the horizontal size of the image in pixels.
Example:	`HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c, IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m;=g&t;=0", "Yahoo"))` This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called `Yahoo Name` on contacts where you can store the contact's Yahoo! Messenger ID.
Tips:	• The `height` and `width` parameters are optional. • Use a text string to replace the `image_url` and `alternate_text` parameters. Surround each text string in quotes. • Use numbers to replace the `height` and `width` parameters. • Add images to your Documents tab if you want to display them elsewhere. For example, store the image of a product in a document folder, copy the URL to the document, and paste that URL in the `image_url` parameter of a formula field on the Products tab. • If you use Internet Explorer, you may need to change your security settings so that it does not display a warning prompt when images use HTTP protocol. See the online help for Internet Explorer for instructions on changing your security settings. • The IMAGE function cannot include the GETSESSIONID function as one of its arguments. • The IMAGE function is available only in formula fields and email templates.

INCLUDE

Description:	Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.
Use:	`{!INCLUDE(source, [inputs])}` and replace `source` with the s-control snippet you want to reference. Replace `inputs` with any information you need to pass to the snippet.

S-Control Example:

Including Header Snippet

```
<html> <body> {! INCLUDE($SControl.Header_Snippet, [title = "My
Title",
                    theme = "modern"])} </body> </html>
```

This example references a snippet that provides a header for a page that you created to display in a Web tab. It displays the page title "My Title." Use the $SControl global variable to reference a custom s-control.

Including Input Parameters

Use the following two examples to see how you can create a reusable snippet and include it in an s-control.

```
<h2
                    class="{!$Request.titleTheme}.title">
{!$Request.titleText}</h2>
```

This snippet requires two input parameters: *titleTheme* and *titleText*. It is a reusable HTML tag that presents a page title and theme based on input parameters. Next, create an s-control that includes this snippet:

```
<html> <head> </head> <body> {!
                    INCLUDE($SControl.Title_Snippet, [titleTheme =
"modern", titleText = "My Sample Title"]) } ... Insert your page
specific content
                    here ... </body> </html>
```

This s-control uses the snippet titled Title_Snippet to display the title of the page "My Sample Title" and modern theme. Replace *Insert your page specific content here* with your own HTML content and use the s-control as the source of a Web tab to create your own pages in Salesforce.

Tips:

- Because this function references an s-control snippet and does not copy it, it always runs the latest content of the s-control snippet. Remember when making a change to your s-control snippet that it affects all INCLUDE functions that refer to it.
- Use the $Request global variable to access any information inside the snippet.
- This function is only available in custom buttons, links, and s-controls.

INCLUDES

Description:	Determines if any value selected in a multi-select picklist field equals a text literal you specify.
Use:	`INCLUDES(`***multiselect_picklist_field, text_literal***`)` and replace *multiselect_picklist_field* with the merge field name for the multi-select picklist; and replace *text_literal* with the multi-select picklist value you want to match in quotes.
Examples:	`INCLUDES(Hobbies__c, "Golf")` returns TRUE if one of the selected values in the `Hobbies` custom multi-select picklist field is Golf.

Tips:

- The *text_literal* expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.
- Salesforce returns an error if any of the following occurs:
  - You do not provide a *text_literal* expression.
  - You provide an empty *text_literal* expression, such as `""` or `" "`.
- Use ISBLANK to determine if a multi-select picklist field is empty.
- Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example:

```
INCLUDES(
   PRIORVALUE(multiselect_picklist_field),
text_literal
)
```

ISBLANK

Description: Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.

Use: `ISBLANK(expression)` and replace *expression* with the expression you want evaluated.

Example:

```
(IF(ISBLANK(Maint_Amount__c), 0, 1) +
 IF(ISBLANK(Services_Amount__c), 0,1) +
  IF(ISBLANK(Discount_Percent__c), 0, 1) +
   IF(ISBLANK(Amount), 0, 1) +
    IF(ISBLANK(Timeline__c), 0, 1)) / 5
```

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A "1" is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the `Treat blank fields as blanks` option under Blank Field Handling while the Advanced Formula subtab is showing.

Tips:

- Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce will continue to support ISNULL, so you do not need to change any existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the BLANKVALUE function to return a specified string if the field does not have a value; use the ISBLANK function if you only want to check if the field has a value.
- If you use this function with a numeric field, the function only returns TRUE if the field has no value and is not configured to treat blank fields as zeroes.
- If you use this function with a picklist, use ISBLANK(TEXT(<picklist>)) to convert the picklist items into a text value.

ISCHANGED

Description:	Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.
Use:	`ISCHANGED(field)` and replace *field* with the name of the field you want to compare.
Validation Rule Example:	The following validation rule prevents users from changing an object name after it has been created: `NOT(ISCHANGED(Name))`. `NOT(AND(ISCHANGED(Priority), ISPICKVAL(Priority, "Low")))` is a validation rule that ensures if a user changes the `Priority` of a case, the new priority cannot be "Low." `NOT(AND(ISCHANGED(CloseDate), OR(MONTH(CloseDate) <> MONTH(TODAY()), YEAR(CloseDate) <> YEAR(TODAY()))),$Profile.Name <> "Sales Manager"))` is a validation rule that prevents a user from changing the `Close Date` of an opportunity to a date outside of the current month and year unless that user has the "Sales Manager" profile. Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Tips:	• This function is available only in: – Assignment rules – Validation rules – Field updates – Workflow rules if the evaluation criteria is set to `Evaluate the rule when a record is: created, and every time it's edited`. – Formula criteria for executing actions in the Process Builder. • Use the NOT function to reverse the return values of TRUE and FALSE. • This function returns FALSE when evaluating any field on a newly created record. • If a text field was previously blank, this function returns TRUE when it contains any value. • For number, percent, or currency fields, this function returns TRUE when: – The field was blank and now contains any value – The field was zero and now is blank – The field was zero and now contains any other value

ISCLONE

Description:	Compares two items and returns TRUE if one item is a clone of the other. Otherwise, returns FALSE.
Use:	`ISCLONE(field1, field2)` and replace the *field* values with the name of the fields to compare for cloning.
Validation Rule Example:	Use the following validation rule to identify a record that's a clone of another record. `AND (ISCLONE(OldAccount, NewAccount)`

Tips:	• Use the NOT function to reverse the return values of TRUE and FALSE.

ISNEW

Description:	Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.
Use:	`ISNEW()`
Validation Rule Example:	Use the following validation rule to prevent users from creating a record with a close date in the past. `AND (ISNEW(), CloseDate < TODAY())` checks if the user is creating a new opportunity and, if so, ensures that the `Close Date` is today or after today. Use this validation rule to ensure users add at least one product to an opportunity after they have created it. `NOT(OR(ISNEW(),HasOpportunityLineItem))` In this example, the validation rule formula displays the following error message when an existing opportunity does not have any products: "You must add products to this opportunity before saving." This does not display an error on the initial save because they cannot add products until after saving the record initially; but it prevents them from resaving or closing an opportunity that does not contain products.
Tips:	• This function is available only in validation rules, field updates, workflow rules, assignment rules, and Process Builder formula criteria. • Use the NOT function to reverse the return values of TRUE and FALSE. • This function always returns FALSE when used in a workflow rule with a time-based trigger. • This function always returns FALSE when used in a field update for an approval action.

ISNULL

Important: Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce will continue to support ISNULL, so you do not need to change any existing formulas.

Description:	Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.
Use:	`ISNULL(`***`expression`***`)` and replace *`expression`* with the expression you want evaluated.

Example:	`(IF(ISNULL(Maint_Amount__c), 0, 1) +`
	`IF(ISNULL(Services_Amount__c), 0,1) +`
	`IF(ISNULL(Discount_Percent__c), 0, 1) +`
	`IF(ISNULL(Amount), 0, 1) +`
	`IF(ISNULL(Timeline__c), 0, 1)) / 5`
	This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that

field. A "1" is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the `Treat blank fields as blanks` option under Blank Field Handling while the Advanced Formula subtab is showing.

Validation Rule Example:

```
AND(ISPICKVAL(StageName, "Closed Won"),
  ISNULL(Project_Start_Date__c))
```

This validation rule makes the `Project Start Date` custom date field conditionally required whenever the stage is "Closed Won."

Tips:

- Text fields are never null, so using this function with a text field always returns false. For example, the formula field `IF(ISNULL(new__c) 1, 0)` is always zero regardless of the value in the `New` field. For text fields, use the ISBLANK function instead.
- Multi-select picklist fields are never null in s-controls, buttons, and email templates, so using this function with a multi-select picklist field in those contexts always returns false.
- Empty date and date/time fields always return true when referenced in ISNULL functions.
- Don't use ISNULL for date/time fields.
- Choose `Treat blank fields as blanks` for your formula when referencing a number, percent, or currency field in an ISNULL function. Choosing `Treat blank fields as zeroes` gives blank fields the value of zero so none of them will be null.
- Merge fields can be handled as blanks, which can affect the results of components like s-controls because they can call this function.
- When using a validation rule to ensure that a number field contains a specific value, use the ISNULL function to include fields that do not contain any value. For example, to validate that a custom field contains a value of '1', use the following validation rule to display an error if the field is blank or any other number:

  ```
  OR(ISNULL(field__c), field__c<>1)
  ```

ISNUMBER

Description: Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.

Use: `ISNUMBER(`***`text`***`)` and replace *`text`* with the merge field name for the text field.

Validation Rule Example:

```
OR(LEN(Bank_Account_Number__c) <> 10,
NOT(ISNUMBER(Bank_Account_Number__c)))
```

This validation rule ensures a custom text field called `Bank Account Number` is a number of 10 digits and is not blank.

Tips:

- This function returns FALSE for blank values.
- The ISNUMBER function is not aware of your locale. For example, `ISNUMBER("123,12")` and `ISNUMBER("1 000")` return FALSE even if the user's locale is "French."
- Chinese, Japanese, Korean, and special characters including a space return FALSE.

- The ISNUMBER function returns TRUE for scientific formatting such as “2E2” or “123.123.”

ISPICKVAL

Description:	Determines if the value of a picklist field is equal to a text literal you specify.
Use:	`ISPICKVAL(`***`picklist_field, text_literal`***`)` and replace *`picklist_field`* with the merge field name for the picklist; replace *`text_literal`* with the picklist value in quotes. *`text_literal`* cannot be a merge field or the result of a function.

Examples:	**Contract Activation**
	`IF(ISPICKVAL(Status, "Activated"), NOW()-ActivatedDate, null)` calculates the number of days since the contract was activated. If the contract status is not “Activated,” this field is blank.
	Commission Amounts
	`IF(ISPICKVAL(StageName, "Closed Won"),`
	`  ROUND(Amount *0.02, 2), 0)`
	This example calculates the commission amount for any opportunity that has a “Closed Won” stage. The value of this field will be the amount times 0.02 for any closed/won opportunity. Open or lost opportunities will have a zero commission value.
	Competitor-Triggered Workflow
	`ISPICKVAL(Stage, “Closed Lost”) && INCLUDES(Competitor__c, “Acme”)`
	In a workflow rule or the Process Builder, this formula configures Salesforce to trigger the associated actions if the `Competitor` multi-select picklist field on a lost business is Acme.

Tips:	• Replace *`picklist_field`* with a custom or standard field of type picklist.
	• Your *`text_literal`* expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.
	• Use CASE functions to determine if a picklist value is equal to a particular value.
	• When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example:
	`ISPICKVAL(PRIORVALUE`
	`(`***`picklist_field`***`),`
	`text_literal``)`

JSENCODE

Description:	Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (’).
Use:	`{!JSENCODE(`***`text`***`)}` and replace *`text`* with the merge field or text string that contains the unsafe JavaScript characters.

Example:	If the merge field `foo__c` contains `<B>Enter the user's name<b>`, `{!JSENCODE(foo__c)}` results in: `\u003CB\u003EEnter the user\'s name\u003C\/b\u003E`
Tips:	This function is only available in custom buttons and links, and in Visualforce.

JSINHTMLENCODE

Description:	Encodes text and merge field values for use in JavaScript inside HTML tags by replacing characters that are reserved in HTML with HTML entity equivalents and inserting escape characters before unsafe JavaScript characters. `JSINHTMLENCODE(`***`someValue`***`)` is a convenience function that is equivalent to `JSENCODE(HTMLENCODE(`***`someValue`***`))`. That is, `JSINHTMLENCODE` first encodes *`someValue`* with `HTMLENCODE`, and then encodes the result with `JSENCODE`.
Use:	`{!JSINHTMLENCODE(`***`text`***`)}` and replace *`text`* with the merge field or text string that contains the unsafe JavaScript characters.
Example:	If the merge field `foo__c` contains `<B>Enter the user's name<b>`, `{!JSINHTMLENCODE(foo__c)}` results in: `<B>Enter the user's name</b>`
Tips:	• This function is only available in custom buttons and links, and in Visualforce.

LEFT

Description:	Returns the specified number of characters from the beginning of a text string.
Use:	`LEFT(`***`text, num_chars`***`)` and replace *`text`* with the field or expression you want returned; replace *`num_chars`* with the number of characters from the left you want returned.
Example:	`TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN__c, 4))` This formula displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this example uses a text custom field called `SSN`.
Tips:	• Reference auto-number fields as text fields in formulas. • If the *`num_chars`* value is less than zero, Salesforce replaces the value with zero.

LEN

Description:	Returns the number of characters in a specified text string.
Use:	`LEN(`***`text`***`)` and replace *`text`* with the field or expression whose length you want returned.
Example:	`LEN(PartNumber__c)` This formula returns the number of characters in a `Product Code` field.

LINKTO

Description: Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce page.

Use: `{!LINKTO(label, target, id, [inputs], [no override]}` and replace *label* with the text for the link, *target* with the URL, and *id* with a reference to the record. Inputs are optional and can include any additional parameters you want to add to the link. The *no override* argument is also optional and defaults to "false." It applies to targets for standard Salesforce pages such as $Action.Account.New. Replace *no override* with "true" when you want to display a standard Salesforce page regardless of whether you have defined an override for it elsewhere.

S-Control Example:

New Account S-Control

```
<html> <body> {!LINKTO("Create a New Account",
                  $Action.Account.New, $ObjectType.Account)}
</body> </html>
```

This example allows users to click a link to create a new account. It is useful in account list views or Web tabs where you want users to create an account directly from that page. Use the $Action global variable to access the new account page in Salesforce.

New Email Window S-Control

```
<html>
<body>
{!LINKTO("Email link",
  "mailto:support@yourcompany.com?subject=Please%20Help")};
</body>
</html>
```

This example launches a new email window addressed to support@yourcompany.com with the subject "Please Help" whenever a user clicks "Mail link."

Link to Another S-Control

```
<html>
<body>
{!LINKTO("Check for duplicates",
  $Scontrol.dedup_account, Account.Id)}
</body>
</html>
```

Use this example to generate a page containing a hyperlink labeled "Check for duplicates." When users click this link, Salesforce runs your custom s-control. This example assumes you have already created a custom s-control to find duplicate accounts and merge their information.

Tips:

- Avoid using this function in an inline s-control if you want it to open in a new window.
- Enclose multiple *inputs* in brackets to indicate they are together:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, [parm1="A",
parm2="B"])}
```

- Set *inputs* to null if you do not have any to pass yet you want to set the *no override* argument:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, null, true)}
```

- When you override the tab home page for a standard or custom tab, set *target* to the "Tab" `$Action` global variable and *id* to the object type. For example, `LINKTO("Accounts Tab", $Action.Account.Tab, $ObjectType.Account)`
- This function is only available in custom buttons, links, and s-controls.

LN

Description:	Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.
Use:	`LN(`***number***`)` and replace *number* with the field or expression for which you want the natural logarithm. Note: the LN function is the inverse of the EXP function.
Example:	`LN(10)` returns the natural logarithm of 10, which is 2.30. `LN(Value__c)` returns the natural logarithm of a custom number field called `Value`.

LOG

Description:	Returns the base 10 logarithm of a number.
Use:	`LOG(`***number***`)` and replace *number* with the field or expression from which you want the base 10 logarithm calculated.
Example:	**Salary** `LOG(Salary__c)` calculates the logarithm of a person's salary. In this example, `Salary` is a custom currency field. **Hydrogen** `-LOG(Hydrogen__c)` calculates the pH and acidity using the LOG function and a custom number field called `Hydrogen`, which represents the concentration of Hydrogen ions in the liquid measured in moles per liter.

LOWER

Description:	Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
Use:	`LOWER(`***text***`, [`***locale***`])` and replace *text* with the field or text you wish to convert to lowercase, and *locale* with the optional two-character ISO language code or five-character locale code, if available.

Example:	**MYCOMPANY.COM** `LOWER("MYCOMPANY.COM")` returns "mycompany.com." **Ticker Symbol** `LOWER(TickerSymbol)` returns the text in `Ticker Symbol` in lower case characters. **Applying Turkish Language Locale Rules** The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the `LOWER()` function with the Turkish language locale, use the Turkish locale code *tr* in the `LOWER()` function as follows: `LOWER(`***`text`***`, "tr")` This ensures that Salesforce does not transform any dotted i in the *`text`* to a dotless I.

LPAD

Description:	Inserts characters you specify to the left-side of a text string.
Use:	`LPAD(`***`text, padded_length`***`[, `***`pad_string`***`])` and replace the variables: • *`text`* is the field or expression you want to insert characters to the left of. • *`padded_length`* is the number of total characters in the text that will be returned. • *`pad_string`* is the character or characters that should be inserted. *`pad_string`* is optional and defaults to a blank space. If the value in *`text`* is longer than *`pad_string`*, *`text`* is truncated to the size of *`padded_length`*.
Example:	**Field Name: Padding** `LPAD(Name, 20)` truncates the `Name` field after 20 characters. For example, if the name is `mycompany.com`, the value returned is "mycompany.com." **My_Company: No Change** `LPAD('my_company.com', 14, 'z')` returns "my_company.com" without change because it has 14 characters. **Field Name Padded with Z** `LPAD(Name, 15, 'z')` returns the name "zmycompany.com." **Field Name: Truncating** `LPAD(Name, 2)` truncates the name after the second character. For example, if the name is `mycompany.com`, the value returned is "my."
Tips:	Leading blank spaces and zeros are omitted.

MAX

Description:	Returns the highest number from a list of numbers.
Use:	`MAX(number, number, ...)` and replace *`number`* with the fields or expressions from which you want to retrieve the highest number.

Example:	**Service Charge**
	`MAX(0.06 * Total_Cost__c, Min_Service_Charge__c)`
	In this example, the formula field calculates a service charge of 6% of the total cost or a minimum service charge, whichever is greater. Note that `Min Service Charge` is a custom currency field with a default value of $15. However, you could make it a formula field if your minimum service charge is always the same amount.
	Book Royalties
	`MAX(0.10 * Pages__c, (Retail_Price__c * 0.07) * Total_Sold__c)`
	This formula determines which amount to pay in royalties for a book. It displays the greater of two amounts: $0.07 for each book sold or $0.10 per page. It assumes you have custom number fields for `Pages` and `Total Sold` and a custom currency field for `Retail Price`.
	Commissions
	`MAX($User.Commission_Percent__c * Price, Price * Account_Discount__c, 100)`
	This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

MID

Description:	Returns the specified number of characters from the middle of a text string given the starting position.
Use:	`MID(text, start_num, num_chars)` and replace *`text`* with the field or expression to use when returning characters; replace *`start_num`* with the number of characters from the left to use as a starting position; replace *`num_chars`* with the total number of characters to return.
Example:	`MID(Division, 3, 4)` returns four characters of the `Division` name beginning with the third character from the left. On a user record, this represents the department code.

MIN

Description:	Returns the lowest number from a list of numbers.
Use:	`MIN(number, number, ...)` and replace *`number`* with the fields or expressions from which you want to retrieve the lowest number.

Example:

401K Matching

```
MIN(250, Contribution__c /2)
```

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or $250, whichever is less. It assumes you have custom currency field for `Contribution`.

Bonus

```
MIN(Gross__c * Bonus_Percent__c,
  Performance__c / Number_of_Employees__c)
```

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have custom number field for `Number of Employees`, a custom percent field for `Bonus Percent`, and currency custom fields for the employee's `Gross` and company's `Performance`.

MOD

Description:	Returns a remainder after a number is divided by a specified divisor.
Use:	`MOD(`***`number, divisor`***`)` and replace *`number`* with the field or expression you want divided; replace *`divisor`* with the number to use as the divisor.

Example:	`MOD(3, 3)` returns 0
	`MOD(4, 3)` returns 1
	`MOD(123, 100)` returns 23
	You may want to prevent users from scheduling meetings on a Saturday or Sunday. Use the following example to apply a validation rule to a custom date field called`My Date`.
	`CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7),`
	`0, 0,`
	`6, 0,`
	`1) = 0`
	This example displays the following error message when the value of `My Date` is not Monday through Friday: "My Date is not a weekday."

MONTH

Description:	Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.
Use:	`MONTH(`***`date`***`)` and replace *`date`* with the field or expression for the date containing the month you want returned.

Example:	**SLA Expiration** `MONTH(SLAExpirationDate__c)` returns the month that your service-level agreement expires. This example uses a custom date field called `SLA Expiration Date`. **Current Month** `MONTH(TODAY())` returns the current month in a number format. For example, the month of February would be the value "2."

NOT

Description:	Returns FALSE for TRUE and TRUE for FALSE.
Use:	`NOT(`***`logical`***`)` and replace *`logical`* with the expression that you want evaluated.
Example:	`IF(NOT(ISPICKVAL(Status, "Closed")), ROUND(NOW()-CreatedDate, 0), null` checks to see if a variable is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the variable is not open, this field is blank.

NOW

Description:	Returns a date/time representing the current moment.
Use:	`NOW()`
Example:	`IF(ISPICKVAL(Status, "Open"), ROUND(NOW()-CreatedDate, 0), null)` This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead is not open, this field is blank.
Tips:	• Do not remove the parentheses. • Keep the parentheses empty. They do not need to contain a value. • Use a date/time field in a NOW function instead of a date field. `Created Date` and `Last Modified Date` are date/time fields whereas `Last Activity Date` is a date field. • Use TODAY if you prefer to use a date field. • Dates and times are always calculated using the user's time zone. • Use addition and subtraction operators with a NOW function and other date/time fields to return a number, representing number of days. For example `NOW() - CreatedDate` calculates the number of days since the created date of a record. In this example, the formula field data type is a number. • Use addition and subtraction operators with a NOW function and numbers to return a date and time. For example `NOW() +5` calculates the date and time five days ahead of now. In this example, the formula field data type is a date/time.

NULLVALUE

Important: Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Salesforce will continue to support NULLVALUE, so you do not need to change existing formulas.

Description:	Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression.
Use:	`NULLVALUE(`***`expression, substitute_expression`***`)` and replace *`expression`* with the expression you want to evaluate; replace *`substitute_expression`* with the value you want to replace any blank values.
Example:	`(NULLVALUE(Sample_Due_Date__c, StartDate +5)` This formula returns the date five days after the start date whenever `Sample Due Date` is blank. `Sample Due Date` is a custom date field.
Tips:	• Avoid using this function with text fields because they are never null even when they are blank. Instead, use the BLANKVALUE function to determine if a text field is blank. • Don't use NULLVALUE for date/time fields. • Choose `Treat blank fields as blanks` for your formula when referencing a number, percent, or currency field in a NULLVALUE function. Choosing `Treat blank fields as zeroes` gives blank fields the value of zero so none of them will be null. • Use the same data type for both the *`expression`* and *`substitute_expression`*.

OR

Description:	Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false. Use this function as an alternative to the operator || (OR).
Use:	`OR(`***`logical1, logical2...`***`)` and replace any number of logical references with the expressions you want evaluated.
Formula Field Example:	`IF(OR(ISPICKVAL(Priority, "High"), ISPICKVAL(Status, "New")), ROUND(NOW()-CreatedDate, 0), null)` This formula returns the number of days a case has been open if the `Status` is new or the `Priority` is high. If the case was opened today, this field displays a zero.
Validation Rule Example:	`OR(Sample_Rate__c < 0, Sample_Rate__c > 0.40)` This validation rule formula displays the following error message when the `Sample Rate` custom field is not between 0 and 40%: "SampleRate cannot exceed 40%."

PARENTGROUPVAL

Description:	This function returns the value of a specified parent grouping. A "parent" grouping is any level above the one containing the formula. You can only use this function in custom summary formulas for reports.
Use:	**Summary and Joined:** `PARENTGROUPVAL(summary_field, grouping_level)` **Matrix:** `PARENTGROUPVAL(summary_field, parent_row_grouping, parent_column_grouping)` Where `summary_field` is the summarized field value, `grouping_level` is the parent level for summary reports, and `parent_row_level` and `parent_column_level` are the parent levels for matrix reports.
Example:	`TOTAL_PRICE:SUM/PARENTGROUPVAL(TOTAL_PRICE:SUM, GRAND_SUMMARY)` This formula calculates, for each product, its relative size compared to the grand total. In this example, the report is a summary of opportunities and their products, grouped by `Product Name`.

PREVGROUPVAL

Description:	This function returns the value of a specified previous grouping. A "previous" grouping is one that comes before the current grouping in the report. Choose the grouping level and increment. The increment is the number of columns or rows before the current summary. The default is 1; the maximum is 12. You can only use this function in custom summary formulas for reports.
Use:	`PREVGROUPVAL(summary_field, grouping_level [, increment])` Where `summary_field` is the name of the grouped row or column, `grouping_level` is the summary level, and `increment` is the number of rows or columns previous.
Example:	`AMOUNT:SUM - PREVGROUPVAL(AMOUNT:SUM, CLOSE_DATE)` This formula calculates, for each month, the difference in amount from the previous month shown in the report. In this example, the report is an opportunity matrix with columns grouped by `Close Date` and rows by `Stage`.

PRIORVALUE

Description:	Returns the previous value of a field.
Use:	`PRIORVALUE(`***`field`***`)`
Validation Rule Example:	The following validation rule prevents users from changing the expected revenue of an opportunity after it is closed: `AND(PRIORVALUE(Amount) > Amount, IsClosed)`.
Tips:	• This function is available only in:

  - Assignment rules
  - Validation rules
  - Field updates
  - Workflow rules if the evaluation criteria is set to `Evaluate the rule when a record is: created, and every time it's edited .`
  - Formula criteria for executing actions in the Process Builder.
- This function does not return default values.
- When users create a new record, this function returns the value of the *`field`* referenced rather than null. For example, if you create an account named "Acme," `PRIORVALUE(Account.Name)` returns `Acme`.
- When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example:

```
ISPICKVAL(PRIORVALUE
(picklist_field),
text_literal)
```

- Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example:

```
INCLUDES(
   PRIORVALUE(multiselect_picklist_field),
text_literal
)
```

REGEX

Description:	Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, it returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.
Use:	`REGEX(text, regex_text)` and replace *`text`* with the text field, and *`regex_text`* with the regular expression you want to match.
Validation Rule Example:	This example ensures that a custom field called `SSN` matches a regular expression representing a valid social security number format of the form 999-99-9999. `NOT( OR( LEN (SSN__c) = 0, REGEX(SSN__c, "[0-9]{3}-[0-9]{2}-[0-9]{4}") ) )`

Tips:

- Regular expression syntax is based on Java Platform SE 6 syntax. However, backslash characters (\) must be changed to double backslashes (\\) because backslash is an escape character in Salesforce.
- The Salesforce regular expression engine matches an entire string as opposed to searching for a match within a string. For example, if you are searching for the name Marc Benioff, use the regular expression, `.*Marc Benioff.*`, to find a match in a string like the following:

  ```
  According to Marc Benioff, the social enterprise increases
  customer success.
  ```

  If you use the regular expression, `Marc Benioff`, the only string that this regular expression will match is:

  ```
  Marc Benioff
  ```

- Capture groups and substitutions are ignored.
- This function is available everywhere formulas exist except formula fields and custom buttons and links.

REQUIRESCRIPT

Description: Returns a script tag with source for a URL you specify. Use this function when referencing the Force.com AJAX Toolkit or other JavaScript toolkits.

Use: `{!REQUIRESCRIPT(`***`url`***`)}` and replace *`url`* with the link for the script that is required.

For the AJAX Toolkit:

```
{!requireScript("/soap/ajax/13.0/connection.js")}
```

Returns:

```
<script src="/soap/ajax/13.0/connection.js"></script>
```

Custom Button Example:

```
{!REQUIRESCRIPT("/soap/ajax/13.0/connection.js")}
var c = new sforce.SObject("Case");
c.id = "{!Case.Id}";
c.Status = "New";
result = sforce.connection.update([c]);
window.location.reload();
```

This example sets the `Status` of a case to "New" whenever a user clicks a custom button from the case detail page. To set this up in your organization, define a custom button for cases that has the following attributes:

- `Display Type` is "Detail Page Button"
- `Behavior` is "Execute JavaScript"
- `Content Source` is "OnClick JavaScript"

Next, paste the content above into your custom button definition and add it to your case page layouts.

Tips:	• Use global variables to access special merge fields for s-controls. • Use this function when creating custom buttons or links where you have set the `Behavior` to "Execute JavaScript" and `Content Source` to "OnClick JavaScript" because the script tag should be outside the OnClick code. • This function is only available for custom buttons and links that have `Content Source` set to "OnClick JavaScript." • When working in Visualforce, use INCLUDESCRIPT instead.

RIGHT

Description:	Returns the specified number of characters from the end of a text string.
Use:	`RIGHT(`***`text, num_chars`***`)` and replace *`text`* with the field or expression you want returned; replace *`num_chars`* with the number of characters from the right you want returned.
Example:	`TRIM(LEFT(LastName, 5))&"-"&TRIM(RIGHT(SSN__c, 4))` displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this assumes you have a text custom field called `SSN`.
Tips:	• Reference auto-number fields as text fields in formulas. • If the *`num_chars`* value is less than zero, Salesforce replaces the value with zero.

ROUND

Description:	Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.
Use:	`ROUND(`***`number, num_digits`***`)` and replace *`number`* with the field or expression you want rounded; replace *`num_digits`* with the number of decimal places you want to consider when rounding.

Example:	`ROUND (1.5, 0) =2`
	`ROUND (1.2345, 0) =1`
	`ROUND (-1.5, 0) =-2`
	`ROUND (225.49823, 2) =225.50`
	Simple Discounting
	`ROUND(Amount-Amount* Discount_Percent__c,2)`
	Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called `Discount Percent`.

Tips:	• Enter zero for *`num_digits`* to round a number to the nearest integer.

- Salesforce automatically rounds numbers based on the decimal places you specify. For example, a custom number field with two decimal places stores 1.50 when you enter 1.49999.
- Salesforce uses the round half-up rounding algorithm. Half-way values are always rounded up. For example, 1.45 is rounded to 1.5. –1.45 is rounded to –1.5.
- The decimal numbers displayed depend on the decimal places you selected when defining the field in the custom field wizard. The *num_digits* represents the number of digits considered when rounding.

RPAD

Description:	Inserts characters that you specify to the right-side of a text string.
Use:	`RPAD(`***`text, padded_length`***`[, '`***`pad_string`***`'])` and replace the variables: • *text* is the field or expression after which you want to insert characters. • *pad_length* is the number of total characters in the text string that will be returned. • *pad_string* is the character or characters that should be inserted. *pad_string* is optional and defaults to a blank space. If the value in *text* is longer than *pad_string*, *text* is truncated to the size of *padded_length*.
Example:	**Field Name: Padding Default** `RPAD(Name, 20)` truncates the `Name` field after 20 characters. For example, if the name is `mycompany.com`, the value returned is "mycompany.com." **My_Company: No Change** `RPAD('my_company.com', 14, 'z')` returns "my_company.com" without change because it has 14 characters. **Field Name: Padding with a Character** `RPAD(Name, 15, 'z')` returns "mycompany.comz" . **Field Name: Truncating** `RPAD(Name, 2)` truncates the name after the second character. For example, if the name is `mycompany.com`, the value returned is "my."
Tips:	Ending blank spaces are omitted.

SQRT

Description:	Returns the positive square root of a given number.
Use:	`SQRT(`***`number`***`)` and replace *number* with the field or expression you want computed into a square root.
Example:	`SQRT(25)` returns the square root of 25, which is 5. **Amplitude**

	`SQRT(Amplitude__c)` returns the square root of a custom number field representing the amplitude of an earthquake.
Tips:	• Calculating the square root of a negative number results in an error on the detail page. • Avoid division by zero errors by including an IF function such as: `IF(Amplitude__c >= 0, SQRT(Amplitude__c), null)`.

SUBSTITUTE

Description:	Substitutes new text for old text in a text string.
Use:	`SUBSTITUTE(`***`text, old_text, new_text`***`)` and replace *`text`* with the field or value for which you want to substitute values, *`old_text`* with the text you want replaced, and *`new_text`* with the text you want to replace the *`old_text`*.
Example:	`SUBSTITUTE(Name, "Coupon", "Discount")` returns the name of an opportunity that contains the term "Coupon" with the opportunity name plus "Discount" wherever the term "Coupon" existed. `SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.")` finds the location of the @ sign in a person's email address to determine the length of text to replace with a "www." as a means of deriving their website address.
Tips:	• Each term provided in quotes is case sensitive. • If the *`old_text`* appears more than once, each occurrence is replaced with the *`new_text`* value provided even when that results in duplicates.

TEXT

Description:	Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, validation rules, formula fields, field updates, and custom buttons and links.
Use:	`TEXT(`***`value`***`)` and replace *`value`* with the field or expression you want to convert to text format. Avoid using any special characters besides a decimal point (period) or minus sign (dash) in this function.
Example:	**Expected Revenue in Text** `TEXT(ExpectedRevenue)` returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the `Expected Revenue` of a campaign is "$200,000," this formula field displays "200000." **Asset ID**

`SerialNumber &"-"& TEXT(Quantity)` returns an asset ID number starting with the serial number and ending with the quantity separated by a dash. The `Serial Number` field is already text but the `Quantity` field is a number, requiring the TEXT function before it.

Use Picklist Values in Math Equations

```
VALUE(LEFT(TEXT(Quantity), 5)) * Unit
```

This formula multiplies the first five numbers of the `Quantity` picklist by the `Unit` numeric field.

Compare Two Picklists

```
IF(TEXT(bug_status) = TEXT(case_status), "Match", "Out of Sync")
```

This formula compares the values of the `bug_status` picklist with values of the `case_status` picklist.

Display Picklist Values From Parent Records

```
TEXT(Account.Industry)
```

This formula field on opportunities shows the industry of the associated account.

Concatenate Picklist Values

```
TEXT(Account.Industry) & " - " & TEXT(Account.SubIndustry__c)
```

This formula field on opportunities shows the industry and subindustry of the associated account.

Validation Rule Examples:

Block the Save of a Closed Opportunity

`CONTAINS(TEXT(Status), "Closed")` returns TRUE if the `Status` picklist contains the value "Closed," such as "Closed Won" and "Closed Lost." This validation rule formula blocks users from saving changes to a closed opportunity.

Use Numeric Functions on Numeric Picklist Values

`VALUE(LEFT(TEXT(Quantity), 5)) * Unit > 10000` multiplies the first five numbers of the `Quantity` picklist by the `Unit` numeric field, and returns TRUE if the result is greater than 10,000.

Directly Compare Two Picklists

`TEXT(bug_status) = TEXT(case_status)` compares the values of the bug_status picklist with values of the case_status picklist, and returns TRUE if they are equal.

Tips:

- The returned text is not formatted with any currency, percent symbols, or commas.
- Values are not sensitive to locale. For example, 24.42 EUR are converted into the number 24.42.
- Percents are returned in the form of a decimal.
- Dates are returned in the form of YYYY-MM-DD, that is, a four-digit year and two-digit month and day.
- Date/time values are returned in the form of YYYY-MM-DD HH:MM:SSZ where YYYY is a four-digit year, MM is a two-digit month, DD is a two-digit day, HH is the two-digit hour, MM are the minutes, SS are the seconds, and Z represents the zero meridian indicating the time is returned in UTC time zone.

- Picklist fields are supported in TEXT functions used in these kinds of formulas: validation rules, approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, formula fields, field updates, and custom buttons and links. In other formulas, use ISPICKVAL or CASE when referencing a picklist field.
- The TEXT function always returns picklist values in your organization's master language, not the language of the current user.

TODAY

Description:	Returns the current date as a date data type.
Use:	`TODAY()`
Example:	`TODAY()-Sample_date_c` calculates how many days in the sample are left.
Validation Rule Example:	`SampleDate < TODAY()` This example ensures that users cannot change the `Sample Date` to any date in the past.
Tips:	• Do not remove the parentheses. • Keep the parentheses empty. They do not need to contain a value. • Use a date field with a TODAY function instead of a date/time field. `Last Activity Date` is a date field whereas `Created Date` and `Last Modified Date` are date/time fields. • See NOW if you prefer to use a date/time field. • Dates and times are always calculated using the user's time zone. • Use addition and subtraction operators with a TODAY function and other date fields to return a number, representing number of days. For example `TODAY()-LastActivityDate` calculates the number of days since the last activity date. In this example, the formula field data type is a number. • Use addition and subtraction operators with a TODAY function and numbers to return a date. For example `TODAY() +5` calculates the date five days ahead of today. In this example, the formula field data type is a date.

TRIM

Description:	Removes the spaces and tabs from the beginning and end of a text string.
Use:	`TRIM(`***`text`***`)` and replace *`text`* with the field or expression you want to trim.
Example:	`TRIM(LEFT(LastName,5))& "-" & RIGHT(FirstName, 1)` returns a network ID for users that contains the first five characters of their last name and first character of their first name separated by a dash.

UPPER

Description:	Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
Use:	`UPPER(text, [locale])` and replace *`text`* with the field or expression you wish to convert to uppercase, and *`locale`* with the optional two-character ISO language code or five-character locale code, if available.
Example:	**MYCOMPANY.COM** `UPPER("mycompany.com")` returns "MYCOMPANY.COM." **MYCOMPANY.COM 123** `UPPER("Mycompany.com 123")` returns "MYCOMPANY.COM 123." **Applying Turkish Language Locale Rules** The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the `UPPER()` function with the Turkish language locale, use the Turkish locale code *tr* in the `UPPER()` function as follows: `UPPER(text, "tr")` This ensures that any dotted i in the *`text`* does not transform to a dotless I.

URLENCODE

Description:	Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in *RFC 3986, Uniform Resource Identifier (URI): Generic Syntax*. For example, blank spaces are replaced with `%20`, and exclamation points are replaced with `%21`.
Use:	`{!URLENCODE(text)}` and replace *`text`* with the merge field or text string that you want to encode.
Example:	If the merge field `foo__c` contains `<B>Mark's page<b>`, `{!URLENCODE(foo_c)}` results in: `%3CB%3EMark%27s%20page%3C%2Fb%3E`
Tips:	• This function is only available in custom buttons and links, and in Visualforce. • Custom buttons and links with URL content sources have separate encoding settings. If you use the URLENCODE function to encode a custom button or link that has an encoding setting specified, Salesforce first encodes the URL according to the custom button or link setting, then encodes the result. For example, if the URL in a custom link contains a space and its encoding setting is UTF8, Salesforce first encodes the space to a plus sign (+), then the URLENCODE function converts the plus sign to its character code, `%2B`. • When you include the standard `Account` field on opportunities (`Opportunity.Account`) in the URLENCODE function, the value of the field is the account ID, not the account name. To encode the account name, create a custom cross-object formula field on opportunities that spans to the account name, and use that field in the URLENCODE function instead of

`Opportunity.Account`. For example, if the cross-object formula is `AccountNameFormula__c`, use the following:

```
http://www.google.com/search?q={!URLENCODE
(Opportunity.AccountNameFormula__c)}
```

URLFOR

Description:	Returns a relative URL for an action, s-control, Visualforce page, or a file in a static resource archive in a Visualforce page.
Use:	`{!URLFOR(target, id, [inputs], [no override])}` and replace *target* with the URL or action, s-control, or static resource merge variable, *id* with a reference to the record, and *inputs* with any optional parameters. The *no override* argument is also optional and defaults to "false." It applies to targets for standard Salesforce pages such as $Action.Account.New. Replace *no override* with "true" when you want to display a standard Salesforce page regardless of whether you have defined an override for it elsewhere. To access a Visualforce page, simply enter the name of your page preceded by an "apex/." For example, if your Visualforce page is named `myTestPage`, you would use `{!URLFOR("/apex/myTestPage")}`.
Visualforce Example:	`<apex:image url="{!URLFOR($Resource.TestZip, 'images/Bluehills.jpg')}" width="50" height="50"/>` In this example, the `<apex:image>` component references a .jpg file contained within a .zip file that has been uploaded as a static resource. When uploaded, the name of the static resource was defined as `TestZip`, and the path to the image within the resource is `images/Bluehills.jpg`.

Tips:	• Use global variables to access special merge fields for actions, s-controls, and static resources.
	• If an input parameter name begins with any character other than a letter or dollar sign ($), enclose it in quotation marks.
	• Enclose multiple *inputs* in brackets to indicate they are together:
	`{!URLFOR($Action.Case.View, Case.Id, [parm1="A", parm2="B"])}`
	• Set *inputs* to null if you do not have any to pass yet you want to set the *no override* argument:
	`{!URLFOR($Action.Case.View, Case.Id, null, true)}`
	• When you override a standard action, that action is no longer available in Salesforce. For example, if you override the new account action, that affects the **New** button on all pages such as the account detail page, account related lists on other detail pages, and the **Create New** drop down list in the sidebar. To override a standard action yet still access it, use the *no override* argument in your s-control to reference that action.

- When you override the tab home page for a standard or custom tab, set `target` to the "Tab" `$Action` global variable and `id` to the object type. For example, `URLFOR($Action.Account.Tab, $ObjectType.Account)`
- This function is only available in custom buttons, links, s-controls, and Visualforce pages.

VALUE

Description:	Converts a text string to a number.
Use:	`VALUE(text)` and replace `text` with the field or expression you want converted into a number.
Example:	**Lead Number** `VALUE(Lead_Number__c)` returns a number for the text value in the auto-number field `Lead Number`. This can be useful if you want to use the `Lead Number` field in a calculation. Note that auto-number fields are actually text fields and must be converted to a number for numeric calculations. **Round Robin Lead Assignment** `MOD(VALUE(Lead_Number__c), 3)` This formula is for a custom formula field named Round_Robin_ID that assigns each lead a value of 0, 1, or 2. This formula uses a custom auto-number field called `Lead Number` that assigns each lead a unique number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example: • Round_Robin_ID = 0 is assigned to Queue A • Round_Robin_ID = 1 is assigned to Queue B • Round_Robin_ID = 2 is assigned to Queue C
Tips:	Make sure the text in a VALUE function does not include any special characters other than a decimal point (period) or minus sign (dash). For example, the formula `VALUE(Text_field__c)` produces these results: • If `Text field` is 123, the result is **123** • If `Text field` is blank, the result is **#Error!** • If `Text field` is $123, the result is **#Error!** • If `Text field` is EUR123, the result is **#Error!**

VLOOKUP

Description:	Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.
Use:	`VLOOKUP(field_to_return, field_on_lookup_object, lookup_value)` and replace `field_to_return` with the field that contains the value you want returned,

	field_on_lookup_object with the field on the related object that contains the value you want to match, and *lookup_value* with the value you want to match.
Validation Rule Example:	This example checks that a billing postal code is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid zip code in the US. If the zip code is not found in the Zip_Code__c object or the billing state does not match the corresponding State_Code__c in the Zip_Code__c object, an error is displayed. `AND( LEN(BillingPostalCode) > 0, OR(BillingCountry = "USA", BillingCountry = "US"), VLOOKUP( $ObjectType.Zip_Code__c.Fields.State_Code__c, $ObjectType.Zip_Code__c.Fields.Name, LEFT(BillingPostalCode,5)) <> BillingState )` **Note:** • Use this example when the billing country is US or USA. • You can download US zip codes in CSV file format from http://zips.sourceforge.net.
Tips:	• The *field_to_return* must be an auto number, roll-up summary, lookup relationship, master-detail relationship, checkbox, date, date/time, email, number, percent, phone, text, text area, or URL field type. • The *field_on_lookup_object* must be the `Record Name` field on a custom object. • The *field_on_lookup_object* and *lookup_value* must be the same data type. • If more than one record matches, the value from the first record is returned. • The value returned must be on a custom object. • You cannot delete the custom field or custom object referenced in this function. • This function is only available in validation rules.

YEAR

Description:	Returns the four-digit year in number format of a given date.
Use:	`YEAR(`***date***`)` and replace *date* with the field or expression that contains the year you want returned.
Example:	`YEAR(TODAY())- YEAR(Initial_Meeting__c)` returns the number of years since your initial meeting with a client. This example uses a custom date field called `Initial Meeting`.

SEE ALSO:

Formula Operators and Functions A–H

Formula Operators and Functions

Using Date and Date/Time Values in Formulas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Date formulas are useful for managing payment deadlines, contract ages, or any other features of your organization that are time or date dependent.

Two data types are used for working with dates: Date and Date/Time. Most values that are used when working with dates are of the Date data type, which store the year, month, and day. Some fields, such as `CreatedDate`, are Date/Time fields, meaning they not only store a date value, but also a time value (stored in GMT but displayed in the users' time zone). Date and Date/Time fields are formatted in the user's locale when viewed in reports and record detail pages.

You can use operations like addition and subtraction on Date and Date/Time values to calculate a future date or elapsed time between two dates. If you subtract one date from another, for example, the resulting value will be the difference between the two initial values in days (Number data type). The same operation between two Date/Time values returns a decimal value indicating the difference in number of days, hours, and minutes.

For example, if the difference between two Date/Time values is 5.52, that means the two values are separated by five days, 12 hours (0.5 of a day), and 28 minutes (0.02 of a day). You can also add numeric values to Dates and Date/Times. For example, the operation `TODAY() + 3` returns three days after today's date. For more information and examples of working with dates, see the list of Sample Date Formulas.

Throughout the examples, the variables *`date`* and *`date/time`* are used in place of actual Date and Date/Time fields or values.

Keep in mind that complex date functions tend to compile to a larger size than text or number formula functions, so you might run into issues with formula compile size. See *Tips for Reducing Formula Size* for help with this problem.

TODAY() and NOW()

The `TODAY()` function returns the current day, month, and year as a Date data type. This function is useful for formulas where you are concerned with how many days have passed since a previous date, the date of a certain number of days in the future, or if you just want to display the current date.

The `NOW()` function returns the Date/Time value of the current moment. It's useful when you are concerned with specific times of day as well as the date.

For details on how to convert between Date values and Date/Time values, see Converting Between Date/Time and Date.

The DATE() Function

The `DATE()` function returns a Date value, given a year, month, and day. Numerical Y/M/D values and the `YEAR()`, `MONTH()`, and `DAY()` functions are valid parameters for `DATE()`. For example `DATE( 2013, 6, 1 )` returns June 6, 2013. Similarly, `DATE( YEAR( TODAY() ), MONTH( TODAY() ) + 3, 1)` returns the Date value of the first day three months from today in the current year, assuming the date is valid (for example, the month falls between 1 and 12).

If the inputted Y/M/D values result in an invalid date, the `DATE()` function returns an error, so error checking is an important part of working with Date values. You can read about methods for handling invalid dates in Sample Date Formulas.

Converting Between Date/Time and Date

Date and Date/Time aren't interchangeable data types, so when you want to perform operations between Date and Date/Time values, you need to convert the values so they are both the same type. Some functions (such as `YEAR()`, `MONTH()`, and `DAY()`) also only work on Date values, so Date/Time values must be converted first.

Use the `DATEVALUE( date/time )` function to return the Date value of a Date/Time. For example, to get the year from a Date/Time, use `YEAR( DATEVALUE( date/time ) )`.

You can convert a Date value to a Date/Time using the `DATETIMEVALUE( date )` function. The time will be set to 12:00 a.m. in Greenwich Mean Time (GMT), and then converted to the time zone of the user viewing the record when it's displayed. For a user located in San Francisco, `DATETIMEVALUE( TODAY() )` returns 5:00 p.m. on the previous day (during Daylight Saving Time) rather than 12:00 a.m. of the current day. See A Note About Date/Time and Time Zones for more information.

Converting Between Date and Text

If you want to include a date as part of a string, wrap the Date value in the `TEXT()` function to convert it to text. For example, if you want to return today's date as text, use:

```
"Today's date is " & TEXT( TODAY() )
```

This returns the date in the format "YYYY-MM-DD" rather than in the locale-dependent format. You can change the format by extracting the day, month, and year from the date first and then recombining them in the format you want. For example:

```
"Today's date is " & TEXT( MONTH( date ) ) & "/" & TEXT( DAY( date ) ) & "/" & TEXT( YEAR(
 date ) ) )
```

You can also convert text to a Date so you can use the string value with your other Date fields and formulas. You'll want your text to be formatted as "YYYY-MM-DD". Use this formula to return the Date value:

```
DATEVALUE( "YYYY-MM-DD" )
```

Converting Between Date/Time and Text

You can include Date/Time values in a string using the `TEXT()` function, but you need to be careful of time zones. For example, consider this formula:

```
"The current date and time is " & TEXT( NOW() )
```

In this formula, `NOW()` is offset to GMT. Normally, `NOW()` would be converted to the user's time zone when viewed, but because it's been converted to text, the conversion won't happen. So if you execute this formula on August 1st at 5:00 PM in San Francisco time (GMT-7), the result is "The current date and time is 2013–08–02 00:00:00Z".

When you convert a Date/Time to text, a "Z" is included at the end to indicate GMT. `TEXT( date/time )` returns "Z" if the field is blank. So if the Date/Time value you're working with might be blank, check for this before converting to text:

```
IF(
 ISBLANK( date/time ),
 "",
 TEXT( date/time )
)
```

To convert a string to a Date/Time value, use `DATETIMEVALUE()` passing in a string in the format "YYYY-MM-DD HH:MM:SS". This method returns the Date/Time value in GMT.

A Note About Date/Time and Time Zones

Date and Date/Time values are stored in GMT. When a record is saved, field values are adjusted from the user's time zone to GMT, and then adjusted back to the viewer's time zone when displayed in record detail pages and reports. With Date conversions this doesn't pose a problem, since converting a Date/Time to a Date results in the same Date value.

When working with Date/Time fields and values, however, the conversion is always done in GMT, not the user's time zone. Subtracting a standard Date/Time field from another isn't a problem because both fields are in the same time zone. When one of the values in the calculation is a conversion from a Text or Date value to a Date/Time value, however, the results are different.

Let's say a San Francisco user enters a value of 12:00 AM on August 2, 2013 in a custom Date/Time field called `Date_Time_c`. This value is stored as 2013–08–02 07:00:00Z, because the time difference in Pacific Daylight Time is GMT-7. At 12:00 p.m. PDT on August 1st, the user views the record and the following formula is run:

```
Date_Time_c - NOW()
```

In the calculation, `NOW()` is 2013–08–01 19:00:00Z, and then subtracted from 2013–08–02 07:00:00Z, to return the expected result of 0.5 (12 hours).

Suppose that instead of NOW(), the formula converts the string "2013–08–01 12:00:00" to a Date/Time value:

```
Date_Time_c - DATETIMEVALUE( "2013-08-01 12:00:00" )
```

In this case, `DATETIMEVALUE( "2013-08-01 12:00:00" )` is 2013–08–01 12:00:00Z, and returns a result of 0.79167, or 19 hours.

There's no way to determine a user's time zone in a formula. If all of your users are in the same time zone, you can adjust the time zone difference by adding or subtracting the time difference between the users' time zone and GMT to your converted values. However, since time zones can be affected by Daylight Saving Time, and the start and end dates for DST are different each year, this is difficult to manage in a formula. We recommend using Apex for transactions that require converting between Date/Time values and Text or Date values.

SEE ALSO:

Tips for Building Formulas

Build a Formula Field

Your custom formula fields require special attributes.

Note: The Getting Started with Formulas help video includes a live demo of these steps.

1. Begin building a formula field the same way you create a custom field. See Create Custom Fields on page 125.
2. Select the data type for the formula. Choose the appropriate data type for your formula based on the output of your calculation. See Formula Data Types on page 232.
3. Choose the number of decimal places for currency, number, or percent data types. This setting is ignored for currency fields in multicurrency organizations. Instead, the `Decimal Places` for your currency setting apply.

   Note: Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and –12.345 becomes –12.34.

4. Click **Next**.
5. Build your formula:
   a. If you are building a formula in the **Advanced Formula** tab or for approvals or rules, such as workflow, validation, assignment, auto-response, or escalation, click **Insert Field**, choose a field, and click **Insert**.

      To create a basic formula that passes specific Salesforce data, select the **Simple Formula** tab, choose the field type in the `Select Field Type` drop-down list, and choose one of the fields listed in the `Insert Field` drop-down list.

      Tip: Build cross-object formulas to span to related objects and reference merge fields on those objects.

   b. To insert an operator, choose the appropriate operator icon from the `Insert Operator` drop-down list.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To view formula field details:
- "View Setup and Configuration"

To create, change, or delete formula fields:
- "Customize Application"

c. Optionally, click the **Advanced Formula** tab to use functions and view additional operators and merge fields. Functions are prebuilt formulas that you can customize with your input parameters.

d. To insert a function, double-click its name in the list, or select it and click **Insert Selected Function**. To filter the list of functions, choose a category from the `Functions` drop-down list. Select a function and click **Help on this function** to view a description and examples of formulas using that function.

e. Consider adding comments to your formula, especially if it is complicated. Comments must begin with a forward slash followed by an asterisk (`/*`), and conclude with an asterisk followed by a forward slash (`*/`).

Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition. For example:

```
AND(
/*competitor field is required, check to see if field is empty */
LEN(Competitor__c) = 0,
/* rule only enforced for ABCD record types */
RecordType.Name = "ABCD Value",
/* checking for any closed status, allows for additional closed picklist values in
the future */
CONTAINS(TEXT(StageName), "Closed")
)
```

You can also use comments to *comment out* sections of your formula when debugging and checking the syntax to locate errors in the formula.

Note:

- Nesting comments causes a syntax error. For example, you cannot save a formula that has the following:

```
/* /* comment */ */
```

- Commenting out a whole formula causes a syntax error.
- Comments count against the character and byte size limits in formulas.

Note: Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula requires more characters, create separate formula fields and reference them in another formula field. The maximum number of displayed characters after an evaluation of a formula expression is 1,300.

6. To check your formula for errors, click **Check Syntax**.
7. Optionally, enter a description of the formula in the `Description` box.
8. If your formula references any number, currency, or percent fields, choose an option for handling blank fields. To give any blank fields a zero value, choose `Treat blank fields as zeros`. To leave these fields blank, choose `Treat blank fields as blanks`.
9. Click **Next**.
10. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
11. Choose the page layouts that should display the field. The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is automatically added to the bottom of the user detail page.
12. Click **Save** to finish or **Save & New** to create more custom fields.

Note: Because formula fields are automatically calculated, they are read-only on record detail pages and do not update last modified date fields. Formula fields are not visible on edit pages.

In account formulas, all business account fields are available as merge fields. However, account fields exclusive to person accounts such as `Birthdate` and `Email` are not available.

IN THIS SECTION:

- Tips for Building Formulas
- What Are Cross-Object Formulas?

SEE ALSO:

- Elements of a Formula
- Merge Fields for Formulas
- Tips for Building Formulas
- Formula Operators and Functions
- Quick Reference Guide: Formula Fields

Tips for Building Formulas

Watch a Demo: Formulas: Tips and Gotchas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Formula fields that a user can see may reference fields that are hidden or read only using field-level security. If the formula field contains sensitive information, use field-level security to hide it.
- You can add activity formula fields to task and event page layouts. Note that a task-related formula field on an event page layout may not be useful. Likewise, event-related formula fields on task page layouts may not be useful.
- To determine if a record is a task or event, use the `IsTask` merge field. For example:

```
IF(IsTask, "This is a task", "This is an event")
```

IN THIS SECTION:

- Tips for Working with Date and Date/Time Formula Fields
- Tips for Using Merge Fields in Formulas
- Tips for Working with Number Formula Fields
- Tips for Working with Picklist and Multi-Select Picklist Formula Fields
- Tips for Referencing Record Types in Formulas
- Tips for Working with Text Formula Fields

SEE ALSO:

- Build a Formula Field
- Common Formula Errors

Tips for Working with Date and Date/Time Formula Fields

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Dates and times are always calculated using the user's time zone.
- Date and date/time fields can't be used interchangeably. The name alone may not indicate if a field is a date or date/time. For example, `Created Date` and `Last Modified Date` are date/time fields whereas `Last Activity Date` is a date field. Use the DATEVALUE function to convert a date/time field into a date field.

  Note: The `Created Date` and `Last Modified Date` fields display only the date, not the date and time.

- Use addition and subtraction operators with date or date/time fields to calculate duration. For example, subtract a date from another date to calculate the number of days between the two. Likewise, you can subtract the date/time from another date/time to get the number of days between the two as a number. See NOW or TODAY for suggested use.
- Use addition and subtraction operators with numbers to return another date or date/time. For example, `{!CreatedDate} + 5` calculates the date and time five days after a record's created date. Note that the expression returns the same data type as the one given; a date field plus or minus a number returns a date, and a date/time field plus or minus a number returns a date/time.
- When calculating dates using fractions, Salesforce ignores any numbers beyond the decimal. For example:

  `TODAY() + 0.7` is the same as `TODAY() + 0`, which is today's date.

  `TODAY() + 1.7` is the same as`TODAY() + 1`, which is tomorrow's date.

  `TODAY() + (-1.8)` is the same as `TODAY() + (-1)`, which is yesterday's date.

- To calculate the value of two fractions first, group them within parentheses. For example:

  `TODAY() + 0.5 + 0.5` is the same as `TODAY() + 0 + 0`, which is today's date.

  `TODAY() + (0.5+0.5)` is the same as `TODAY() + 1`, which is tomorrow's date.

- Years can't be zero and must be between -4713 and 9999.

SEE ALSO:

Tips for Building Formulas

Tips for Using Merge Fields in Formulas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Delegated administrators need to have access to custom objects to access the objects' merge fields from formulas.
- In account formulas, all business account fields are available as merge fields. However, account fields exclusive to person accounts such as `Birthdate` and `Email` are not available.
- You can't use formula fields that include related object merge fields in roll-up summary fields.
- Formulas and roll-up summary fields can't reference fields on external objects.
- Using `RecordType.Id` can make your formula less readable; when you do use it, write in-line comments into the formula to clarify.
- To determine if a record is a task or event, use the `IsTask` merge field. For example:

```
IF(IsTask, "This is a task", "This is an event")
```

- To reference the unique identifier for your Salesforce organization in a formula, insert the `$Organization.Id` merge field. This merge field can display anywhere formula fields can except in reports.
- Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

Use the values "Read," "Edit," and "None" in a formula when referencing:

- $UserRole.CaseAccessForAccountOwner
- $UserRole.OpportunityAccessForAccountOwner
- CaseAccessLevel (on Territory)
- OpportunityAccessLevel (on Territory)

Use the values "Read," "Edit," and "All" in a formula when referencing:

- AccountAccessLevel (on Territory)

- If you create a contacts formula field that references account merge fields, that field can be included in contact page layouts but should not be included in person accounts page layouts. The formula field will display a value of `#Error` on the person accounts page.

SEE ALSO:

Tips for Building Formulas

Tips for Working with Number Formula Fields

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Use the decimal version of a percent when working with percent fields in formulas. For example, `IF(Probability =1...)` for 100% probability or `IF(Probability =0.9...)` for 90% probability.
- Reference auto-number fields as text fields in formulas.
- The output of your formula must be less than 19 digits.
- Formulas can contain a mix of numbers, percents, and currencies as in this example: `AnnualRevenue / NumberOfEmployees`.
- Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and –12.345 becomes –12.34.

SEE ALSO:

Tips for Building Formulas

Tips for Working with Picklist and Multi-Select Picklist Formula Fields

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- You can use special picklist fields in your formulas, such as `IsEscalated` for cases and `IsWon` for opportunities.
- Picklist fields can only be used in these functions:
  - ISPICKVAL—Compares the value of a picklist to a single value.
  - CASE—Compares the value of a picklist to multiple values.
  - TEXT—Converts a picklist value into a text value so that you can work with the value in functions that support text value, such as CONTAINS. (Available in only flow formula resources, formula fields, validation rules, and workflow field updates.)
- The TEXT function always returns picklist values in your organization's master language, not the language of the current user.
- Multi-select picklist fields can only be used in these functions:
  - INCLUDES
  - ISBLANK

  - ISNULL
  - ISCHANGED (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it's edited**)
  - PRIORVALUE (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it's edited**)

SEE ALSO:

Tips for Building Formulas

Tips for Referencing Record Types in Formulas

Reference record types in formulas if you want different workflow rules, validation rules, and lookup filters to apply to different record types. For example, you can:

- Create a workflow rule on accounts that emails different teams based on the account record type the user selects when creating the account.
- Create a validation rule on opportunities that allows only members of the North American sales team to save opportunities with the Domestic record type.

When possible, use `RecordTypeId` instead of `RecordType.Name` to reference a specific record type. While `RecordType.Name` makes a formula more readable, you must update the formula if the name of the record type changes, whereas the ID of a record type never changes. Also, `RecordType.Name` requires a cross-object reference to the record type, while `RecordTypeId` doesn't. However, if you are deploying formulas across organizations (for example, between sandbox and production), use `RecordType.Name` because IDs are not the same across organizations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Avoid using `$RecordType` in formulas, except in default value formulas. Instead, use the `RecordType` merge field (for example, `Account.RecordType.Name`) or the `RecordTypeId` field on the object.

SEE ALSO:

Tips for Building Formulas

Tips for Working with Text Formula Fields

- Before using the HYPERLINK function, consider the differences between hyperlinks and custom links.
  - Hyperlink formula fields are just like other custom fields that you can display in list views and reports.
  - Custom links display on detail pages in a predefined section; hyperlink formula fields can display on a detail page wherever you specify.
  - Using custom links, you can specify display properties such as window position and opening in a separate popup position; hyperlink formula fields open in a new browser window by default or you can specify a different target window or frame.
  - Your formulas can reference custom links. Before deleting a custom link, make sure it is not referenced in a formula field.
  - Hyperlink formula fields that contain relative URLs to Salesforce pages, such as `/rpt/reportwizard.jsp`, can be added to list views, reports, and related lists. However, use a complete URL, including the server name and `https://`, in your hyperlink formula before adding it to a search layout. Formula fields aren't available in search result layouts.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- To insert text in your formula field, surround the text with quotation marks. For example, to display “CASE: 123,” use this formula `"CASE: "& CaseNumber__c`.
- Use the backslash (\) character before a quote or backslash to insert it as a literal value in your output. For example, `"Trouble\\Case \"Ticket\": "` in your formula displays Trouble\Case "Ticket": on detail pages.

SEE ALSO:

Tips for Building Formulas

What Are Cross-Object Formulas?

Cross-object formulas are formulas that span two related objects and reference merge fields on those objects. Cross-object formulas can reference merge fields from a master (“parent”) object if an object is on the detail side of a master-detail relationship. Cross-object formulas also work with lookup relationships.

You can reference fields from objects that are up to 10 relationships away. Cross-object formulas are available anywhere formulas are used except when creating default values.

Note: If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don’t have access to that object record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don’t have access to the account record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

IN THIS SECTION:

Building Cross-Object Formulas in the Simple Formula Tab

Build Cross-Object Formulas in the Advanced Formula Tab

Tips for Building Cross-Object Formulas

SEE ALSO:

Build a Formula Field

Building Cross-Object Formulas in the Simple Formula Tab

To create a cross-object formula when building a formula in the Simple Formula tab, enter the relationship names of the objects to which you are spanning followed by the field you want to reference. Separate the relationship names of each object and the field with periods.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To create or change cross-object formulas:

- “Customize Application”

Example: For example, enter `Contact.Account.Name` to reference the `Account Name` for a contact associated with a case in a formula field on the Case object. Be sure to use the relationship names of the objects, not the labels. Although the relationship name is often the same as the object name, it is technically the field name of the relationship field.

To reference the parent account name from Account object, the syntax is `Parent.Name`, not `Account.Name`. When referencing a custom object, add two underscores and the letter *r* to its name. For example, `Position__r.title__c` references the `Job Title` field (`title__c`) on a Position custom object.

Note: If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don't have access to that object record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don't have access to the account record.

SEE ALSO:

Build Cross-Object Formulas in the Advanced Formula Tab

What Are Cross-Object Formulas?

Build Cross-Object Formulas in the Advanced Formula Tab

To create a cross-object formula when building a formula in the Advanced Formula tab or for approvals or rules, such as workflow, validation, assignment, auto-response, or escalation rules, click **Insert Field**, then click the related object to list its fields. Related objects are denoted by a ">" sign.

Note: If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don't have access to that object record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don't have access to the account record.

Example: The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected; however, in list views and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure the formula always returns the intended result. For example:

```
IF
     (OR
          (LastModifiedBy.Profile.Name = "Standard User",
LastModifiedBy.Profile.Name = "PT2"),
     "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

SEE ALSO:

Building Cross-Object Formulas in the Simple Formula Tab

What Are Cross-Object Formulas?

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To create or change cross-object formulas:

- "Customize Application"

Tips for Building Cross-Object Formulas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Cross-object formulas that reference currency fields convert the value to the currency of the record that contains the formula. If the referenced currency field is from a custom setting, the field value isn't converted to the record's currency.
- Salesforce allows a maximum of 10 unique relationships per object in cross-object formulas. The limit is cumulative across all formula fields, rules, and lookup filters. For example, if two different formulas on opportunities reference two different fields of an associated account, only one unique relationship exists (from opportunities to accounts).
- You can't reference cross-object formulas in roll-up summary fields.
- In cross-object formulas, you can't reference merge fields for objects related to activities. For example, merge fields for contacts and accounts aren't available in task and event formulas.

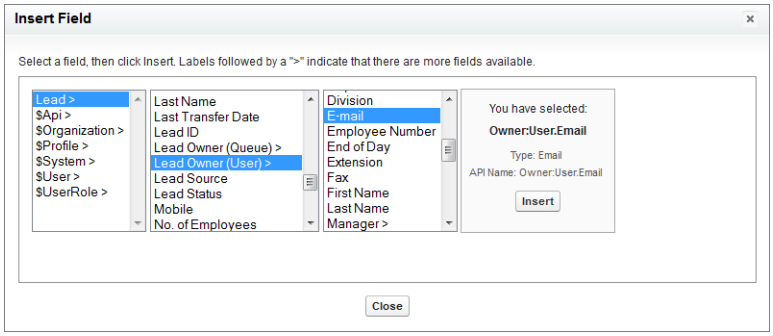
Using the Owner Field

Some objects support different object types for the Owner field, such as a User, Queue, or Calendar. On objects that support this behavior, when creating a cross-object formula using Owner, you must be explicit about the owner type you're referencing.

For example, if you need owner email and you don't use queues, your formula would be `Owner:User.Email`. If you do use queues, your formula could be

```
IF( ISBLANK( Owner:User.Id ), Owner:Queue.QueueEmail, Owner:User.Email )
```

Here's how you would select Owner object fields on a Lead in the Advanced formula tab:



Note:

- Owner references aren't supported in Visualforce pages. For example, on a page with Case as a controller, you can't include `{!Case.Owner:User.FirstName}`. However, you can include an existing spanning formula on a Visualforce page. For example, if you have a custom text formula `MyFormula__c` on a Case with value `Owner:User.FirstName`, you can include `{!Case.MyFormula__c}` on your Visualforce page.
- Owner references aren't supported on the Queue object. For example, you can't reference `Owner:Queue.Owner.Email`.
- If your formula has `Owner:User.fieldname` and `Owner:Queue.fieldname`, both of these count against the limit of 10 unique relationships per object in cross-object formulas.
- On objects that don't support Queues, User is implicit when referencing Owner. Your formula should be `Owner.fieldname`, not `Owner:User.fieldname`.

Using `Profile.Name`

The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected; however, in list views and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure the formula always returns the intended result. For example:

```
IF
     (OR
          (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.Profile.Name =
"PT2"),
     "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

SEE ALSO:

Build Cross-Object Formulas in the Advanced Formula Tab

What Are Cross-Object Formulas?

Formula Field Limits and Restrictions

Before you create formula fields, be aware of their limits and limitations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- Formula fields have these limits.
  - Character limit—Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula needs more characters, create separate formula fields and reference them in another formula field.

    Note: The maximum number of displayed characters after an evaluation of a formula expression is 1,300.
  - Save size limit—Formula fields can't exceed 4,000 bytes when saved. The save size is different from the number of characters if you use multi-byte characters in your formula.
  - Compile size limit—Formula fields can't exceed 5,000 bytes when compiled. The compile size is the size of the formula (in bytes) including all of the fields, values, and formulas it references. There is no direct correlation between the compile size and the character limit. Some functions, such as TEXT, DATEVALUE, DATETIMEVALUE, and DATE significantly increase the compile size.

  Tip: For tips on how to rework your formulas to avoid these limits, see *Tips for Reducing Formula Size*
- Default value formulas for a record type can only reference fields for that record type. However, formula fields and formulas for approvals or rules for a record type can reference fields for that record type as well as any records that are related through a lookup or master-detail relationship. For example, a formula for a validation rule on opportunities can reference merge fields for accounts and campaigns as well as opportunities, and a formula field on accounts can reference fields for cases.
- You can't use long text area, encrypted, or `Description` fields in formulas.
- The value of a field can't depend on another formula that references it.
- You can't delete fields referenced in formulas. Remove the field from the formula before deleting it.

- Campaign statistic fields can't be referenced in formulas for field updates, approval processes, workflow rules, or validation rules, but can be referenced in custom formula fields.

SEE ALSO:

Tips for Building Formulas

Build a Formula Field

Examples of Advanced Formula Fields

Review examples of formula fields for various types of apps that you can use and modify for your own purposes.

This document contains custom formula samples for the following topics. For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

USER PERMISSIONS

To view formula field details:

- "View Setup and Configuration"

To create, change, or delete formula fields:

- "Customize Application"

IN THIS SECTION:

Sample Account Management Formulas

Sample Account Media Service Formulas

Sample Case Management Formulas

Sample Commission Calculations Formulas

Sample Contact Management Formulas

Sample Data Categorization Formulas

Sample Date Formulas

Sample Discounting Formulas

Sample Employee Services Formulas

Sample Expense Tracking Formulas

Sample Financial Calculations Formulas

Sample Image Link Formulas

Sample Integration Link Formulas

Sample Lead Management Formulas

Sample Metrics Formulas

Sample Opportunity Management Formulas

Sample Pricing Formulas

Sample Scoring Calculations Formulas

SEE ALSO:

Formulas: How Do I ... ?

Tips for Building Formulas

Quick Reference Guide: Formula Fields

Sample Account Management Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Account Rating

This formula evaluates `Annual Revenue`, `Billing Country`, and `Type`, and assigns a value of "Hot," "Warm," or "Cold."

```
IF (AND (AnnualRevenue > 10000000,
CONTAINS (CASE (BillingCountry, "United States", "US", "America", "US", "USA", "US", "NA"),
 "US")),
IF(ISPICKVAL(Type, "Manufacturing Partner"), "Hot",
IF(OR (ISPICKVAL (Type, "Channel Partner/Reseller"),
ISPICKVAL(Type, "Installation Partner")), "Warm", "Cold")),
"Cold")
```

In addition, you can reference this Account Rating formula field from the contact object using cross-object formulas.

```
Account.Account_Rating__c
```

Account Region

This formula returns a text value of "North," "South," "East," "West," or "Central" based on the `Billing State/Province` of the account.

```
IF(ISBLANK(BillingState), "None",
IF(CONTAINS("AK:AZ:CA:HA:NV:NM:OR:UT:WA", BillingState), "West",
IF(CONTAINS("CO:ID:MT:KS:OK:TX:WY", BillingState), "Central",
IF(CONTAINS("CT:ME:MA:NH:NY:PA:RI:VT", BillingState), "East",
IF(CONTAINS("AL:AR:DC:DE:FL:GA:KY:LA:MD:MS:NC:NJ:SC:TN:VA:WV", BillingState), "South",
IF(CONTAINS("IL:IN:IA:MI:MN:MO:NE:ND:OH:SD:WI", BillingState), "North", "Other"))))))
```

Contract Aging

This formula calculates the number of days since a contract with an account was activated. If the contract `Status` is not "Activated," this field is blank.

```
IF(ISPICKVAL(Contract_Status__c, "Activated"),
NOW() - Contract_Activated_Date__c, null)
```

Sample Account Media Service Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

BBC™ News Search

This formula creates a link to a BBC news search site based on the `Account Name`.

```
HYPERLINK(
"http://www.bbc.co.uk/search/news/?q="&Name,
"BBC News")
```

Bloomberg™ News Search

This formula creates a link to an account's ticker symbol on the Bloomberg website.

```
HYPERLINK(
"http://www.bloomberg.com/markets/symbolsearch?query="&TickerSymbol,
"Bloomberg News")
```

CNN™ News Search

This formula creates a link to a CNN news search site using the `Account Name`.

```
HYPERLINK(
"http://http://www.cnn.com/search/?query="&Name,
"CNN News")
```

MarketWatch™ Search

This formula creates a link to an account's ticker symbol on the Marketwatch.com website.

```
HYPERLINK(
"http://www.marketwatch.com/investing/stock/"&TickerSymbol,
"Marketwatch")
```

Google™ Search

This formula creates a link to a Google search site using the `Account Name`.

```
HYPERLINK(
"http://www.google.com/#q="&Name,
"Google")
```

Google News Search

This formula creates a link to a Google news search site using the `Account Name`.

```
HYPERLINK(
"http://news.google.com/news/search?en&q="&Name,
"Google News")
```

Yahoo!™ Search

This formula creates a link to a Yahoo! search site using the `Account Name`.

```
HYPERLINK(
"http://search.yahoo.com/search?p="&Name,
"Yahoo Search")
```

Yahoo! News Search

This formula creates a link to a Yahoo! news search site using the `Account Name`.

```
HYPERLINK(
"http://news.search.yahoo.com/search/news?p="&Name,
"Yahoo News")
```

Sample Case Management Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: Salesforce Classic

Available in all editions

Autodial

This formula creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace `"servername"` and `"call"` with the name of your dialing tool and the command it uses to dial. The merge field, `Id`, inserts the identifier for the contact, lead, or account record. The first `Phone` merge field tells the dialing tool what number to call and the last `Phone` merge field uses the value of the `Phone` field as the linkable text the user clicks to dial.

```
HYPERLINK("http://servername/call?id=" & Id & "&phone=" &
                      Phone, Phone)
```

Case Categorization

This formula displays a text value of "RED," "YELLOW," or "GREEN," depending on the value of a case age custom text field.

```
IF(DaysOpen__c > 20, "RED",
IF(DaysOpen__c > 10, "YELLOW",
"GREEN") )
```

Case Data Completeness Tracking

This formula calculates the percentage of specific custom fields that contain data. The formula checks the values of two custom number fields: `Problem Num` and `Severity Num`. If the fields are empty, the formula returns the value "0." The formula returns a value of "1" for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

```
(IF(ISBLANK(Problem_Num__c), 0, 1) + IF(ISBLANK(Severity_Num__c ), 0,1)) * 50
```

Suggested Agent Prompts

This formula prompts an agent with cross-sell offers based on past purchases.

```
CASE(Product_Purch__c,
"Printer", "Extra toner cartridges", "Camera", "Memory cards",
"Special of the day")
```

Suggested Offers

This formula suggests a product based on the support history for a computer reseller. When the `Problem` custom field matches a field, the formula field returns a suggestion.

```
CASE(Problem__c,
"Memory", "Suggest new memory cards", "Hard Drive failure", "Suggest new hard drive with
tape backup",
"")
```

Sample Commission Calculations Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Commission Amounts for Opportunities

The following is a simple formula where commission is based on a flat 2% of the opportunity `Amount`.

```
IF(ISPICKVAL(StageName, "Closed Won"),
  ROUND(Amount *0.02, 2), 0)
```

This example calculates the commission amount for any opportunity that has a “Closed Won” stage. The value of this field will be the amount times 0.02 for any closed/won opportunity. Open or lost opportunities will have a zero commission value.

Commission Deal Size

This formula calculates a commission rate based on deal size, returning a 9% commission rate for deals over 100,000 and an 8% commission rate for smaller deals.

```
IF(Amount > 100000, 0.09, 0.08 )
```

Commission Greater Than or Equal To

This formula assigns the “YES” value with a commission greater than or equal to one million. Note, this is a text formula field that uses a custom currency field called `Commission`.

```
IF(Commission__c >=
                    1000000, "YES", "NO")
```

Commission Maximum

This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

```
MAX($User.Commission_Percent__c * Price,
  Price * Account_Discount__c, 100)
```

Sample Contact Management Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Contact's Account Discount Percent

This percent formula displays the account's `Discount Percent` field on the contacts page.

```
Account.Discount_Percent__c
```

Contact's Account Name

This formula displays the standard `Account Name` field on the contacts page.

```
Account.Name
```

Contact's Account Phone

This formula displays the standard `Account Phone` field on the contacts page.

```
Account.Phone
```

Contact's Account Rating

Use this formula to display the `Account Rating` field on the contacts page.

```
CASE(Account.Rating, "Hot", "Hot", "Warm", "Warm", "Cold", "Cold", "Not Rated")
```

Contact's Account Website

This formula displays the standard `Account Website` field on the contacts page.

```
Account.Website
```

If the account website URL is long, use the HYPERLINK function to display a label such as “Click Here” instead of the URL. For example:

```
IF(Account.Website="", "",
 IF(
  OR(LEFT(Account.Website, 7) = "http://",LEFT(Account.Website, 8) = "https://"),
  HYPERLINK( Account.Website , "Click Here" ),
  HYPERLINK( "http://" & Account.Website , "Click Here" )
```

```
  )
)
```

This formula also adds the necessary "http://" or "https://" before a URL if neither were included in the URL field.

Contact's LinkedIn™ Profile

You can configure a link that appears on your contacts' profile page that sends you to their LinkedIn profile. To do so:

1. From the object management settings for contacts, go to Buttons, Links, and Actions.
2. Click **New Button or Link**.
3. Enter a `Label` for this link, like *`LinkedInLink`*.
4. Enter this formula in the content box:

   ```
   http://www.linkedin.com/search/fpsearch?type=people&keywords
   ={!Contact.FirstName}+{!Contact.LastName}
   ```

5. Click **Save**.

Remember to add this link to the Contact page layout in order for it to show up.

Contact Identification Numbering

This formula displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this example uses a text custom field called `SSN`.

```
TRIM(LEFT(LastName, 5)) &
                    "-" & TRIM(RIGHT(SSN__c, 4))
```

Contact Preferred Phone

This formula displays the contact's preferred contact method in a contact related list—work phone, home phone, or mobile phone—based on a selected option in a `Preferred Phone` custom picklist.

```
CASE(Preferred_Phone__c,
"Work", "w. " & Phone,
"Home", "h. " & HomePhone,
"Mobile", "m. " & MobilePhone,
"No Preferred Phone")
```

Contact Priority

This formula assesses the importance of a contact based on the account rating and the contact's title. If the account rating is `Hot` or the title starts with `Executive`, then the priority is high (P1). If the account rating is `Warm` or the title starts with `VP` then the priority is medium (P2), and if the account rating is `Cold` then the priority is low (P3).

```
IF(OR(ISPICKVAL(Account.Rating, "Hot"), CONTAINS(Title, "Executive")), "P1",

IF(OR(ISPICKVAL(Account.Rating, "Warm"), CONTAINS(Title, "VP")), "P2",

IF(ISPICKVAL(Account.Rating, "Cold"), "P3",

"P3")
```

```
)
)
```

Contact Yahoo! ID

This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called `Yahoo Name` on contacts where you can store the contact's Yahoo! Messenger ID.

```
HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,
                    IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c &
                    "&m;=g&t;=0", "Yahoo"))
```

Dynamic Address Formatting

This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country.

```
CASE(ShippingCountry,
"USA",
  ShippingStreet & BR() &
  ShippingCity & ",
  " & ShippingState & " " &
  ShippingPostalCode & BR()
  & ShippingCountry,
"France",
  ShippingStreet & BR() &
  ShippingPostalCode & " " &
  ShippingCity & BR() &
  ShippingCountry, "etc")
```

Telephone Country Code

This formula determines the telephone country code of a contact based on the `Mailing Country` of the mailing address.

```
CASE(MailingCountry,
"USA", "1",
"Canada", "1",
"France", "33",
"UK", "44",
"Australia", "61",
"Japan", "81",
"?")
```

Unformatted Phone Number

This formula removes the parentheses and dash characters from North American phone numbers. This is necessary for some auto-dialer software.

```
IF(Country_Code__c = "1", MID( Phone ,2, 3) & MID(Phone,7,3) & MID(Phone,11,4), Phone)
```

Sample Data Categorization Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Deal Size Large and Small

This formula displays "Large Deal" for deals over one million dollars or "Small Deal" for deals under one million dollars.

```
IF(Sales_Price__c > 1000000,
"Large Deal",
"Small Deal")
```

Deal Size Small

This formula displays "Small" if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

```
IF(AND(Price<1,Quantity<1),"Small", null)
```

Product Categorization

This formula checks the content of a custom text field named `Product_Type` and returns "Parts" for any product with the word "part" in it. Otherwise, it returns "Service." Note that the values are case sensitive, so if a `Product_Type` field contains the text "Part" or "PART," this formula returns "Services."

```
IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")
```

Sample Date Formulas

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Finding the Day, Month, or Year from a Date

Use the functions `DAY( date )`, `MONTH( date )`, and `YEAR( date )` to return their respective numerical values. Replace ***date*** with a value of type Date (e.g. `TODAY()`).

To use these functions with Date/Time values, first convert them to a date with the `DATEVALUE()` function. For example, `DAY( DATEVALUE( date/time ))`.

Finding Out if a Year Is a Leap Year

This formula determines whether or not a year is a leap year. A year is only a leap year if it's divisible by 400, or if it's divisible by four but NOT by 100.

```
OR(
  MOD( YEAR( date ), 400 ) = 0,
  AND(
   MOD( YEAR( date ), 4 ) = 0,
    MOD( YEAR( date ), 100 ) != 0
  )
)
```

Finding Which Quarter a Date Is In

For standard quarters, you can determine which quarter a date falls in using this formula. This formula returns the number of the quarter in which *date* falls (1–4) by dividing the current month by three (the number of months in each quarter) and taking the ceiling.

```
CEILING( MONTH ( date ) / 3 )
```

The formula for shifted quarters is similar, but shifts the month of the date by the number of months between January and the first quarter of the fiscal year. The example below illustrates how you can find a date's quarter if Q1 starts in February instead of January.

```
CEILING( ( MONTH ( date ) - 1 ) / 3)
```

If you want to check whether a date is in the current quarter, add a check to compare the date's year and quarter with `TODAY()`'s year and quarter.

```
 AND(
   CEILING( MONTH( date ) / 3 ) = CEILING( MONTH( TODAY() ) / 3 ),
   YEAR( date ) = YEAR( TODAY() )
)
```

Finding the Week of the Year a Date Is In

To find the number of a date's week of the year, use this formula:

```
IF(
  CEILING( ( date - DATE( YEAR( date ), 1, 1) + 1) / 7) > 52,
  52,
  CEILING( ( date - DATE( YEAR( date ), 1, 1) + 1) / 7)
)
```

You can find the current week by determining how many days there have been in the current year and dividing that value by 7. The `IF()` statement ensures that the week number the formula returns doesn't exceed 52. So if the given date is December 31 of the given year, the formula returns 52, even though it's more than 52 weeks after the week of January.

Finding Whether Two Dates Are in the Same Month

To determine whether two Dates fall in the same month, say for a validation rule to determine whether an opportunity Close Date is in the current month, use this formula:

```
AND(
  MONTH( date_1 ) == MONTH( date_2 ),
  YEAR( date_1 ) == YEAR( date_2 )
)
```

Finding the Last Day of the Month

The easiest way to find the last day of a month is to find the first day of the next month and subtract a day.

```
IF(
  MONTH( date ) = 12,
  DATE( YEAR( date ), 12, 31 ),
  DATE( YEAR( date ), MONTH ( date ) + 1, 1 ) - 1
)
```

Displaying the Month as a String Instead of a Number

To return the month as a text string instead of a number, use:

```
CASE(
  MONTH( date ),
  1, "January",
  2, "February",
  3, "March",
  4, "April",
  5, "May",
  6, "June",
  7, "July",
  8, "August",
  9, "September",
  10, "October",
  11, "November",
  "December"
)
```

If your organization uses multiple languages, you can replace the names of the month with a custom label:

```
CASE(
  MONTH( date ),
  1, $Label.Month_of_Year_1,
  2, $Label.Month_of_Year_2,
  3, $Label.Month_of_Year_3,
  4, $Label.Month_of_Year_4,
  5, $Label.Month_of_Year_5,
  6, $Label.Month_of_Year_6,
  7, $Label.Month_of_Year_7,
  8, $Label.Month_of_Year_8,
  9, $Label.Month_of_Year_9,
  10, $Label.Month_of_Year_10,
  11, $Label.Month_of_Year_11,
  $Label.Month_of_Year_12
)
```

Finding and Displaying the Day of the Week From a Date

To find the day of the week from a Date value, use a known Sunday (e.g. January 7, 1900) and subtract it from the date (e.g. `TODAY()`) to get the difference in days. The `MOD()` function finds the remainder of this result when divided by 7 to give the numerical value of the day of the week between 0 (Sunday) and 6 (Saturday). The formula below finds the result and then returns the text name of that day.

```
CASE(
  MOD( date - DATE( 1900, 1, 7 ), 7 ),
  0, "Sunday",
  1, "Monday",
  2, "Tuesday",
  3, "Wednesday",
  4, "Thursday",
  5, "Friday",
  "Saturday"
)
```

Note that this formula only works for dates after 01/07/1900. If you're working with older dates, use the same process with any Sunday prior to your earliest date (e.g. 01/05/1800).

You can also adjust this formula if your week starts on a different day. For example, if your week starts on Monday, you can use January 8, 1900 in your condition. The new formula looks like this:

```
CASE(
  MOD( date - DATE( 1900, 1, 8 ), 7 ),
  0, "Monday",
  1, "Tuesday",
  2, "Wednesday",
  3, "Thursday",
  4, "Friday",
  5, "Saturday",
  "Sunday"
)
```

Like the formula for getting the name of the month, if your organization uses multiple languages, you can replace the names of the day of the week with a variable like `$Label.Day_of_Week_1`, etc.

Finding the Next Day of the Week After a Date

To find the date of the next occurrence of a particular day of the week following a given Date, get the difference in the number of days of the week between a *date* and a *day_of_week*, a number 0–6 where 0 = Sunday and 6 = Saturday. By adding this difference to the current date, you'll find the date of the *day_of_week*. The `IF()` statement in this formula handles cases where the *day_of_week* is prior to the day of the week of the *date* value (e.g. *date* is a Thursday and *day_of_week* is a Monday) by adding 7 to the difference.

```
date + ( day_of_week - MOD( date - DATE( 1900, 1, 7 ), 7 ) )
+
IF(
  MOD( date - DATE( 1900, 1, 7 ), 7 ) >= day_of_week,
  7,
  0
)
```

You can substitute either a constant or another field in for the *day_of_week* value based on your needs.

Finding the Number of Days Between Two Dates

To find the number of days between two dates, *date_1* and *date_2*, subtract the earlier date from the later date: ***date_1*** − ***date_2***

You can alter this slightly if you want to determine a date a certain number of days in the past. For example, say you want a formula to return true if some date field is more than 30 days prior to the current date and false otherwise. This formula does just that:

```
TODAY() - 30 > date
```

Finding the Number of Business Days Between Two Dates

Calculating how many business days passed between two dates is slightly more complex than calculating total elapsed days. The basic strategy is to choose a reference Monday from the past and find out how many full weeks and any additional portion of a week have

passed between the reference date and the date you're examining. These values are multiplied by five (for a five-day work week) and then the difference between them is taken to calculate business days.

```
(5 * ( FLOOR( ( date_1 - DATE( 1900, 1, 8) ) / 7 ) ) + MIN( 5, MOD( date_1 - DATE( 1900,
 1, 8), 7 ) ) )
-
(5 * ( FLOOR( ( date_2 - DATE( 1900, 1, 8) ) / 7 ) ) + MIN( 5, MOD( date_2 - DATE( 1900,
 1, 8), 7 ) ) )
```

In this formula, *date_1* is the more recent date and *date_2* is the earlier date. If your work week runs shorter or longer than five days, replace all fives in the formula with the length of your week.

Adding Days, Months, and Years to a Date

If you want to add a certain number of days to a date, add that number to the date directly. For example, to add five days to a date, the formula is `date + 5`.

Adding years to a date is fairly simple, but you do need to check that the future date is valid. That is, adding five years to February 29 (a leap year) results in an invalid date. The following formula adds *num_years* to *date* by checking if the date is February 29 and if the future date is not in a leap year. If these conditions hold true, the formula returns March 1 in the future year. Otherwise, the formula sets the Date to the same month and day *num_years* in the future.

```
IF(
  AND(
    MONTH( date ) = 2,
    DAY( date ) = 29,
    NOT(
      OR(
        MOD( YEAR( date ), 400 ) = 0,
        AND(
          MOD( YEAR( date ), 4 ) = 0,
          MOD( YEAR( date ), 100 ) != 0
        )
      )
    )
  ),
  DATE( YEAR( date ) + num_years, 3, 1),
  DATE( YEAR( date ) + num_years, MONTH( date ), DAY( date ) )
)
```

Adding months to a date is slightly more complicated as months vary in length and the cycle of months restart with each year. Therefore, a valid day in one month (January 31) might not be valid in another month (February 31). A simple solution is to approximate each month's length as 365/12 days:

```
date + ( ( 365 / 12 ) * Number_months )
```

While this formula is a good estimate, it doesn't return an exact date. For example, if you add two months to April 30 using this method, the formula will return June 29 instead of June 30. Returning an exact date depends on your organization's preference. For example, when you add one month to January 31, should it return February 28 (the last day of the next month) or March 2 (30 days after January 31)?

This formula does the following:

- Returns March 1 if the future month is a February and the day is greater than 28. This portion of the formula performs the same for both leap and non-leap years.

- Returns the first day of the next month if the future month is April, June, September, or November and the day is greater than 30.
- Otherwise, it returns the correct date in the future month.

This example formula adds two months to a given date. You can modify the conditions on this formula if you prefer different behaviors for dates at the end of the month.

```
DATE(
  YEAR( date ) + FLOOR( ( MONTH ( date ) + 2 - 1 ) / 12 ),
  MOD( MONTH ( date ) + 2 - 1 +
    IF( DAY ( date ) > CASE( MOD( MONTH( date ) + 2 - 1, 12 ) + 1,
      2, 28,
      4, 30,
      6, 30,
      9, 30,
      11, 30,
      31 ), 1, 0 ), 12 ) + 1,
    IF( DAY( date ) > CASE( MOD( MONTH( date ) + 2 - 1, 12 ) + 1,
      2, 28,
      4, 30,
      6, 30,
      9, 30,
      11, 30,
      31 ),
    1, DAY( date )
  )
)
```

If you're using these formulas for expiration dates, you might want to subtract a day from the return value to make sure that some action is completed *before* the calculated date.

Adding Business Days to a Date

This formula finds three business days from a given `date`.

```
CASE(
  MOD( date - DATE( 1900, 1, 7 ), 7 ),
  3, date + 2 + 3,
  4, date + 2 + 3,
  5, date + 2 + 3,
  6, date + 1 + 3,
  date + 3
)
```

This formula finds the day of the week of the `date` field value. If the date is a Wednesday, Thursday, or Friday, the formula adds five calendar days (two weekend days, three weekdays) to the date to account for the weekend. If `date` is a Saturday, you need four additional calendar days. For any other day of the week (Sunday — Tuesday), simply add three days. You can easily modify this formula to add more or less business days. The tip for getting the day of the week might be useful if you need to adjust this formula.

Finding the Hour, Minute, or Second from a Date/Time

To get the hour, minute, and second from a Date/Time field as a numerical value, use the following formulas where `TZoffset` is the difference between the user's time zone and GMT. For hour in 24–hour format:

```
VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
```

For hour in 12–hour format:

```
IF(
  OR(
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12
  ),
  12,
  VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
   -
   IF(
     VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
     0,
     12
   )
)
```

For minutes:

```
VALUE( MID( TEXT( date/time - TZoffset ), 15, 2 ) )
```

For seconds:

```
VALUE( MID( TEXT( date/time - TZoffset ), 18, 2 ) )
```

And, to get "AM" or "PM" as a string, use:

```
IF(
  VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
  "AM",
  "PM"
)
```

To return the time as a string in "HH:MM:SS A/PM" format, use the following formula:

```
IF(
  OR(
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12
  ),
  "12",
  TEXT( VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
   -
   IF(
     VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
     0,
     12
   )
  )
)
& ":" &
MID( TEXT( date/time - TZoffset ), 15, 2 )
& ":" &
MID( TEXT( date/time - TZoffset ), 18, 2 )
& " " &
IF(
```

```
  VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
  "AM",
  "PM"
)
```

When working with time in formula fields, you need to consider the time difference between your organization and GMT. See A Note About Date/Time and Time Zones on page 287 for help understanding the time zone offset used in this formula.

Finding the Elapsed Time Between Date/Times

To find the difference between two Date values as a number, subtract one from the other like so: ***date_1*** – ***date_2*** to return the difference in days.

Finding the elapsed time between two Date/Time values is slightly more complex. This formula converts the difference between two Date/Time values, ***datetime_1*** and ***datetime_2***, to days, hours, and minutes.

```
IF(
  datetime_1 - datetime_2 > 0 ,
  TEXT( FLOOR( datetime_1 - datetime_2 ) ) & " days "
  & TEXT( FLOOR( MOD( (datetime_1 - datetime_2 ) * 24, 24 ) ) ) & " hours "
  & TEXT( ROUND( MOD( (datetime_1 - datetime_2 ) * 24 * 60, 60 ), 0 ) ) & " minutes",
  ""
)
```

Finding the Number of Business Hours Between Two Date/Times

The formula for finding business hours between two Date/Time values expands on the formula for finding elapsed business days. It works on the same principle of using a reference Date/Time, in this case 1/8/1900 at 16:00 GMT (9 a.m. PDT), and then finding your Dates' respective distances from that reference. The formula rounds the value it finds to the nearest hour and assumes an 8–hour, 9 a.m. – 5 p.m. work day.

```
ROUND( 8 * (
    ( 5 * FLOOR( ( DATEVALUE( date/time_1 ) - DATE( 1900, 1, 8) ) / 7) +
     MIN(5,
      MOD( DATEVALUE( date/time_1 ) - DATE( 1900, 1, 8), 7) +
      MIN( 1, 24 / 8 * ( MOD( date/time_1 - DATETIMEVALUE( '1900-01-08 16:00:00' ), 1 ) )
 )
     )
   )
 -
    ( 5 * FLOOR( ( DATEVALUE( date/time_2 ) - DATE( 1900, 1, 8) ) / 7) +
      MIN( 5,
       MOD( DATEVALUE( date/time_2 ) - DATE( 1996, 1, 1), 7 ) +
       MIN( 1, 24 / 8 * ( MOD( date/time_2 - DATETIMEVALUE( '1900-01-08 16:00:00' ), 1) )
 )
     )
   )
  ),
0 )
```

You can change the eights in the formula to account for a longer or shorter work day. If you live in a different time zone or your work day doesn't start at 9:00 a.m., change the reference time to the start of your work day in GMT. See A Note About Date/Time and Time Zones for more information.

SEE ALSO:

- Using Date and Date/Time Values in Formulas
- Examples of Advanced Formula Fields
- Tips for Building Formulas

Sample Discounting Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Maintenance and Services Discount

This formula field uses two custom currency fields: `Maintenance Amount` and `Services Amount`. It displays "Discounted" on an opportunity if its maintenance amount and services amount do not equal the opportunity `Amount` standard field value. Otherwise, it displays "Full Price."

```
IF(Maintenance_Amount__c + Services_Amount__c <> Amount,
"Discounted",
"Full Price")
```

Opportunity Discount Amount

This formula calculates the difference of the product `Amount` less the `Discount Amount`. Note that `Discount Amount` is a custom currency field.

```
Amount -
                    Discount_Amount__c
```

Opportunity Discount Rounded

Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called `Discount Percent`.

```
ROUND(Amount-Amount* Discount_Percent__c,2)
```

Opportunity Discount with Approval

This formula adds a "Discount Approved" checkbox to an opportunity. It uses conditional logic to check the value of the approval flag before calculating the commission.

```
IF(Discount_Approved__c, ROUND(Amount - Amount * DiscountPercent__c, 2), Amount)
```

Sample Employee Services Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Bonus Calculation

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have custom number field for `Number of Employees`, a custom percent field for `Bonus Percent`, and currency custom fields for the employee's `Gross` and company's `Performance`.

```
MIN(Gross__c * Bonus_Percent__c,
  Performance__c / Number_of_Employees__c)
```

Employee 401K

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or $250, whichever is less. It assumes you have custom currency field for `Contribution`.

```
MIN(250, Contribution__c /2)
```

Hours Worked Per Week

This formula uses a custom tab to enable time tracking of hours worked per day. It uses a formula field to sum the hours per week.

```
MonHours__c + TuesHours__c + WedsHours__c + ThursHours__c + FriHours__c
```

Total Pay Amount

This formula determines total pay by calculating regular hours multiplied by a regular pay rate, plus overtime hours multiplied by an overtime pay rate.

```
Total Pay =
IF(Total_Hours__c <= 40, Total_Hours__c * Hourly_Rate__c,
40 * Hourly_Rate__c +
(Total_Hours__c - 40) * Overtime_Rate__c)
```

Sample Expense Tracking Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Expense Identifier

This formula displays the text “Expense-” followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

```
"Expense-" &
                    Trip_Name__c & "-" & ExpenseNum__c
```

Mileage Calculation

This formula calculates mileage expenses for visiting a customer site at 35 cents a mile.

```
Miles_Driven__c * 0.35
```

Sample Financial Calculations Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Compound Interest

This formula calculates the interest, you will have after T years, compounded M times per year.

```
Principal__c * ( 1 + Rate__c / M ) ^ ( T * M) )
```

Compound Interest Continuous

This formula calculates the interest that will have accumulated after T years, if continuously compounded.

```
Principal__c * EXP(Rate__c * T)
```

Consultant Cost

This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of $1200 per day. Note that `Consulting Days` is a custom field.

```
Consulting_Days__c *
                    1200
```

Gross Margin

This formula provides a simple calculation of gross margin. In this formula example, `Total Sales` and `Cost of Goods Sold` are custom currency fields.

```
Total_Sales__c - Cost_of_Goods_Sold__c
```

Gross Margin Percent

This formula calculates the gross margin based on a margin percent.

```
Margin_percent__c * Items_Sold__c * Price_item__c
```

Payment Due Indicator

This formula returns the date five days after the contract start date whenever `Payment Due Date` is blank. `Payment Due Date` is a custom date field.

```
(BLANKVALUE(Payment_Due_Date__c, StartDate +5)
```

Payment Status

This formula determines if the payment due date is past and the payment status is "UNPAID." If so, it returns the text "PAYMENT OVERDUE" and if not, it leaves the field blank. This example uses a custom date field called `Payment Due Date` and a text custom field called `Payment Status` on contracts.

```
IF(
AND(Payment_Due_Date__c < TODAY(),
ISPICKVAL(Payment_Status__c, "UNPAID")),
"PAYMENT OVERDUE",
null )
```

Sample Image Link Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Yahoo! Instant Messenger™ Image

This formula displays an image that indicates whether a contact or user is currently logged in to Yahoo! Instant Messenger. Clicking the image launches the Yahoo! Instant Messenger window. This formula uses a custom text field called `Yahoo Name` to store the contact or user's Yahoo! ID.

```
IF(ISBLANK(Yahoo_Name__c),"", HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,
IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m=g&t=0", " ")))
```

Flags for Case Priority

This formula displays a green, yellow, or red flag image to indicate case priority.

```
IMAGE(
CASE( Priority,
"Low", "/img/samples/flag_green.gif",
"Medium", "/img/samples/flag_yellow.gif",
"High", "/img/samples/flag_red.gif",
"/s.gif"),
"Priority Flag")
```

Color Squares for Case Age

This formula displays a 30 x 30 pixel image of a red, yellow, or green, depending on the value of a `Case Age` custom number field.

```
IF( Case_Age__c > 20,
IMAGE("/img/samples/color_red.gif", "red", 30, 30),
IF( Case_Age__c > 10,
IMAGE("/img/samples/color_yellow.gif", "yellow", 30, 30),
IMAGE("/img/samples/color_green.gif", "green", 30, 30)
))
```

Traffic Lights for Status

This formula displays a green, yellow, or red traffic light images to indicate status, using a custom picklist field called `Project Status`. Use this formula in list views and reports to create a "Status Summary" dashboard view.

```
IMAGE(
CASE(Project_Status__c,
"Green", "/img/samples/light_green.gif",
"Yellow", "/img/samples/light_yellow.gif",
"Red", "/img/samples/light_red.gif",
"/s.gif"),
"status color")
```

Stars for Ratings

This formula displays a set of one to five stars to indicate a rating or score.

```
IMAGE(
CASE(Rating__c,
"1", "/img/samples/stars_100.gif",
"2", "/img/samples/stars_200.gif",
"3", "/img/samples/stars_300.gif",
"4", "/img/samples/stars_400.gif",
"5", "/img/samples/stars_500.gif",
"/img/samples/stars_000.gif"),
"rating")
```

Consumer Reports™—Style Colored Circles for Ratings

This formula displays a colored circle to indicate a rating on a scale of one to five, where solid red is one, half red is two, black outline is three, half black is four, and solid black is five.

```
IMAGE(
CASE(Rating__c,
"1", "/img/samples/rating1.gif",
"2", "/img/samples/rating2.gif",
"3", "/img/samples/rating3.gif",
"4", "/img/samples/rating4.gif",
"5", "/img/samples/rating5.gif",
"/s.gif"),
"rating")
```

Horizontal Bars to Indicate Scoring

This formula displays a horizontal color bar (green on a white background) of a length that is proportional to a numeric score. In this example, the maximum length of the bar is 200 pixels.

```
IMAGE("/img/samples/color_green.gif", "green", 15, Industry_Score__c * 2) &
IMAGE("/s.gif", "white", 15,
200 - (Industry_Score__c * 2))
```

Sample Integration Link Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Application API Link

This formula creates a link to an application outside Salesforce, passing the parameters so that it can connect to Salesforce via the SOAP API and create the necessary event.

```
HYPERLINK ("https://www.myintegration.com?sId=" & GETSESSIONID() & "?&rowID=" & Name &
"action=CreateTask","Create a Meeting Request")
```

Important: `$Api.Session_ID` and `GETSESSIONID()` return the same value, an identifier for the current session in the current context. This context varies depending on where the global variable or function is evaluated. For example, if you use either in a custom formula field, and that field is displayed on a standard page layout in Salesforce Classic, the referenced session will be a basic Salesforce session. That same field (or the underlying variable or formula result), when used in a Visualforce page, references a Visualforce session instead.

Session contexts are based on the domain of the request. That is, the session context changes whenever you cross a hostname boundary, such as from `.salesforce.com` to `.visual.force.com` or `.lightning.force.com`.

Session identifiers from different contexts, and the sessions themselves, are different. When you transition between contexts, the old session is replaced by the new one, and the old session is no longer valid. The session ID also changes at this time.

Normally Salesforce transparently handles session hand-off between contexts, but if you're passing the session ID around yourself, be aware that you might need to re-access `$Api.Session_ID` or `GETSESSIONID()` from the new context to ensure a valid session ID.

Note also that not all sessions are created equal. In particular, sessions obtained in a Lightning Experience context have reduced privileges, and don't have API access. You can't use these session IDs to make API calls.

Shipment Tracking Integration

This formula creates a link to FedEx, UPS, or DHL shipment tracking websites, depending on the value of a `Shipping Method` custom picklist field. Note that the parameters shown in this example for FedEx, UPS, and DHL websites are illustrative and do not represent the correct parameters for all situations.

```
CASE(Shipping_Method__c,
"Fedex",
HYPERLINK("http://www.fedex.com/Tracking?ascend_header=1&clienttype
=dotcom&cntry_code=us&language=english&tracknumbers= "& tracking_id__c,"Track"),
"UPS",
HYPERLINK("http://wwwapps.ups.com/WebTracking/processInputRequest?HTMLVersion
=5.0&sort_by=status&loc=en_US&InquiryNumber1= "& tracking_id__c & "&track.x=32&track.y=7",
 "Track") ,
"DHL",
HYPERLINK("http://track.dhl-usa.com/TrackByNbr.asp?ShipmentNumber=" &
tracking_id__c,"Track"), "")
```

Skype™ Auto Dialer Integration

This formula creates a linkable phone number field that automatically dials the phone number via the Skype VOIP phone application. It requires installation of the Skype application (a third-party product not provided by Salesforce) on your desktop.

```
HYPERLINK("callto://+" & Country_Code__c & Phone_Unformatted__c, Phone)
```

Sample Lead Management Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Lead Aging (for open leads)

This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead is not open, this field is blank.

```
IF(ISPICKVAL(Status,
                    "Open"), ROUND(NOW()-CreatedDate, 0), null)
```

Lead Data Completeness

This formula calculates the percent of certain lead fields that your sales personnel enter. The formula field checks the values of two custom number fields: `Phone` and `Email`. If the fields are empty, the formula returns the value "0." The formula returns a value of "1" for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

```
(IF(Phone = "", 0, 1) + IF(Email = "", 0, 1) ) * 50
```

Lead Numbering

This formula returns a number value for the text value in the auto-number field `Lead Number`. This can be useful if you want to use the `Lead Number` field in a calculation, such as round-robin or other routing purposes. Note that auto-number fields are text fields and must be converted to a number for numeric calculations.

```
VALUE(Lead_Number__c)
```

Round-Robin Assignment of Cases or Leads

The following formula example for leads assumes you have three lead queues and you want to assign an equal number of incoming leads to each queue. You can also assign cases using a similar formula.

```
MOD(VALUE(Lead_Number__c),
                  3)
```

This formula is for a custom formula field named Round_Robin_ID that assigns each lead a value of 0, 1, or 2. This formula uses a custom auto-number field called `Lead Number` that assigns each lead a unique number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example:

- Round_Robin_ID = 0 is assigned to Queue A
- Round_Robin_ID = 1 is assigned to Queue B
- Round_Robin_ID = 2 is assigned to Queue C

Sample Metrics Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Temperature Conversion

This formula converts Celsius degrees to Fahrenheit.

```
1.8 * degrees_celsius__c + 32
```

Unit of Measure Conversion

This formula converts kilometers to miles.

```
Miles__c/.621371192
```

Sample Opportunity Management Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Expected Product Revenue

This formula calculates total revenue from multiple products, each with a different probability of closing.

```
ProductA_probability__c * ProductA_revenue__c + ProductB_probability__c * ProductB_revenue__c
```

Maintenance Calculation

This formula calculates maintenance fees as 20% of license fees per year. `Maintenance Years` is a custom field on opportunities.

```
Amount * Maint_Years__c * 0.2
```

Monthly Subscription-Based Calculated Amounts

This formula calculates an opportunity amount based on a monthly subscription rate multiplied by the subscription period.

```
Monthly_Amount__c * Subscription_Months__c
```

Monthly Value

This formula divides total yearly value by 12 months.

```
Total_value__c / 12
```

Opportunity Additional Costs

This formula calculates the sum of the product `Amount`, maintenance amount, and services fees. Note that `Maint amount` and `Service Fees` are custom currency fields.

```
Amount + Maint_Amount__c +
                    Services_Amount__c
```

Opportunity Categorization

This formula uses conditional logic to populate an `Opportunity category` text field, based on the value of the `Amount` standard field. Opportunities with amounts less than $1500 are "Category 1," those between $1500 and $10000 are "Category 2," and the rest are "Category 3." This example uses nested IF statements.

```
IF(Amount < 1500, "Category 1", IF(Amount > 10000, "Category 3", "Category 2"))
```

Opportunity Data Completeness

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A "1" is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the `Treat blank fields as blanks` option under Blank Field Handling while the Advanced Formula subtab is showing.

```
(IF(ISBLANK(Maint_Amount__c), 0, 1) +
 IF(ISBLANK(Services_Amount__c), 0,1) +
  IF(ISBLANK(Discount_Percent__c), 0, 1) +
   IF(ISBLANK(Amount), 0, 1) +
    IF(ISBLANK(Timeline__c), 0, 1)) / 5
```

Opportunity Expected License Revenue

This formula calculates expected revenue for licenses based on probability of closing.

```
Expected_rev_licenses__c * Probability
```

Opportunity Revenue Text Display

This formula returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the `Expected Revenue` of a campaign is "$200,000," this formula field displays "200000."

```
TEXT(ExpectedRevenue)
```

Opportunity Total Deal Size

This formula calculates the sum of maintenance and services amounts.

```
Amount + Maint_Amount__c + Services_Amount__c
```

Opportunity Total Price Based on Units

This formula generates proposal pricing based on unit price and total volume.

```
Unit_price__c * Volume__c * 20
```

Professional Services Calculation

This formula estimates professional service fees at an average loaded rate of $1200 per day. `Consulting Days` is a custom field on opportunities.

```
Consulting_Days__c * 1200
```

Stage-Based Sales Document Selection

This formula Identifies a relevant document in the Documents tab based on opportunity `Stage`. Use document IDs in the form of "00l30000000j7AO."

```
CASE(StageName,
"Prospecting", "Insert 1st Document ID",
"Qualification", "Insert 2nd Document ID",
"Needs Analysis", "Insert 3rd Document ID",
"Value Proposition", ...
)
)
```

Sales Coach

This formula creates a hyperlink that opens a stage-specific document stored in the Documents tab. It uses the previously defined custom formula field that identifies a document based on opportunity `Stage`. See Stage-Based Sales Document Selection on page 323.

```
HYPERLINK("/servlet/servlet.FileDownload?file=" & Relevant_Document__c, "View Document in
 New Window")
```

Shipping Cost by Weight

This formula calculates postal charges based on weight.

```
package_weight__c * cost_lb__c
```

Shipping Cost Percentage

This formula calculates shipping cost as a fraction of `total amount`.

```
Ship_cost__c / total_amount__c
```

Tiered Commission Rates

This formula calculates the 2% commission amount of an opportunity that has a probability of 100%. All other opportunities will have a commission value of zero.

```
IF(Probability = 1,
ROUND(Amount * 0.02, 2),
0)
```

Total Contract Value from Recurring and Non-Recurring Revenue

This formula calculates both recurring and non-recurring revenue streams over the lifetime of a contract.

```
Non_Recurring_Revenue__c + Contract_Length_Months__c * Recurring_Revenue__c
```

Sample Pricing Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Total Amount

This formula calculates a total amount based on unit pricing and total units.

```
Unit_price__c * Total_units__c
```

User Pricing

This formula calculates a price per user license.

```
Total_license_rev__c / Number_user_licenses__c
```

Sample Scoring Calculations Formulas

For details about using the functions included in these samples, see Formula Operators and Functions on page 234.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Lead Scoring

This formula scores leads, providing a higher score for phone calls than website requests.

```
CASE(LeadSource, "Phone", 2, "Web", 1, 0)
```

Here's a formula that scores a lead based on his or her rating:

```
CASE(1, IF(ISPICKVAL(Rating, "Hot"), 1, 0), 3, IF(ISPICKVAL(Rating, "Warm"), 1, 0), 2,
IF(ISPICKVAL(Rating, "Cold"), 1, 0), 1))
```

Customer Success Scoring

This formula uses a simple scoring algorithm to rank customers a high score for positive survey results in Salesforce.

```
Survey_Question_1__c * 5 + Survey_Question_2__c *2
```

Formulas: How Do I ... ?

Common Math Calculations

- Add numbers?
- Convert text into a number?
- Divide numbers?
- Multiply numbers?
- Round numbers?
- Subtract numbers?

Common Text Functions

- Check if a field contains specified text?
- Check if a picklist field contains a specified value?
- Combine first and last names?
- Convert numbers into text?
- Create a hyperlink field?

Advanced Formulas

- Calculate Commission Amounts for Opportunities?
- Set Up Round-Robin Assignment of Cases or Leads?
- Set Up Opportunity Discount Rounded?

Custom Summary Formulas for Reports

- Calculate the sum of all leads that have **Email Opt Out** and **Do Not Call** fields selected?
- Calculate the difference of all **Amount** fields and all **Discounted Amount** fields on opportunities?
- Calculate the average age of all opportunities?
- Calculate what percent of all opportunities are closed won?
- Calculate the number of active Salesforce users to the 2nd power for administration?
- Calculate the duration of all activities (minutes) times the number of records per 24 hours?
- Calculate the average percent margin on a product-by-product level across many opportunities?
- Calculate the percentage of one product compared to all products in closed opportunities?
- Calculate the change in revenue from opportunities between months?

Cross-Object Formulas

- Display a `Percent` field from a parent object?
- Display a text field from a parent object?
- Display a phone number field from a parent object?
- Display a picklist field from a parent object?
- Display a URL field from a parent object?

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Some How Do I's are not relevant to **Database.com**

USER PERMISSIONS

To view formula field details:

- "View Setup and Configuration"

To create, change, or delete formula fields:

- "Customize Application"

Common Formula Errors

Review common errors that can occur with formulas and how to fix them.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

- "#Error!" displays for a formula field whenever an error occurs while calculating the value of a formula. To resolve the error, check your formula.
  - Is the formula dividing by zero? If so, check if the denominator of your expression is zero and provide an alternative value. For example, the following campaign formula field is blank if the number of opportunities is zero:

    ```
    IF(NumberOfOpportunities > 0,
      NumberOfWonOpportunities / NumberOfOpportunities, null)
    ```

  - Is the formula calculating a value larger than the maximum value of the current type? If so, you can append L to numeric values to make them Long so the intermediate products will be Long and no overflow occurs. For example, the following example shows how to correctly compute the amount of milliseconds in a year by multiplying Long numeric values.

    ```
    Long MillsPerYear = 365L * 24L * 60L * 60L * 1000L;
                                                        Long ExpectedValue =
    31536000000L;

    System.assertEquals(MillsPerYear, ExpectedValue);
    ```

  - Is the formula calculating the square root of a negative number? If so, use an IF function similar to the one above to check if the value is a positive number.
  - Is the formula calculating the LOG of a negative number? If so, use an IF function similar to the one above to make sure that the number is positive.
  - Is the formula using the VALUE function with text that contains special characters? For examples of special characters, see Formula Operators and Functions on page 234.
  - Make sure the formula does not contain a HYPERLINK function within a text function, such as `LEFT( HYPERLINK("http://MYCOMPANY.ORG ", "MYCOMPANY ") , 5)`.
  - Is the formula disabled or referencing a disabled formula field? Salesforce disables formula fields when they are deleted and they remain disabled after they are restored. To enable disabled formula fields, edit and save the field. For more information on deleted custom fields and restoring them, see Manage Deleted Custom Fields on page 144.
- "#Too Big!" displays if your formula output is over 18 digits. When this happens, check your formula for calculations that could result in more than 18 digits. Avoid multiplying large numbers, raising a large number to a power, or dividing by a very small number.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one should be returned. For example, `CASE(Field__c,"Partner", "P", "Customer", "C", LEFT(Field__c, -5))` returns an error even if the value of the field is "Partner" or "Customer" because the last statement is illogical.
- Prevent division by zero errors by including an IF function that determines if the value of a field is zero. For example, `IF(Field__c =0,0, 25/Field__c)`.

Merge Fields

Merge Fields Overview

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record. For example, you can place a merge field in an email template so that the greeting includes the recipient's name rather than a generic "Hello!". You can use merge fields within custom formula fields, s-controls, custom links, custom buttons, Visualforce pages, and when you create email or mail merge templates.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available merge fields vary according to which Salesforce edition you have.

Merge field names are determined when you create a new custom field or object. `Field Name` is automatically populated based on what you type into `Field Label`. You can customize this field if you want, but keep in mind that the name must:

- Only use underscores and alphanumeric characters
- Begin with a letter and end with a letter
- Not include spaces
- Not contain two consecutive underscores

Important: Ensure that the custom field name and label are unique for that object.

- If a standard and custom field have identical names or labels, the merge field displays the custom field value.
- If two custom fields have identical names or labels, the merge field may display an unexpected value.

If you create a field label called *`Email`* and a standard field labeled `Email` already exists, the merge field may be unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *`Email2`*.

To find the merge field name for an object or field in Salesforce, visit the object or field's detail page and refer to `Field Name`.

To incorporate merge fields, use the editor in the respective feature. Salesforce provides valid merge fields in each editor for all related standard and custom objects. If you're using the Connect for Office Word add-in to create mail merge templates, you'll see a complete list of valid merge fields to insert.

Syntax

A merge field's syntax can vary depending on where you're using the field. To make sure you're using the correct syntax, select merge fields from the drop-down list in the editor where you're using the merge field.

Custom objects and fields are always appended with `__c` when referenced. Field labels are preceded by the object, and all spaces are converted to underscores. For example, `Account.CreatedDate` references the `Created Date` standard field for the account object.

In standard relationships, the name of the relationship is the master object. For example, you can reference the account name from a contact validation rule using `Account.Name`; you can reference the phone number of the account creator from an opportunity product formula field using `Opportunity.Account.CreatedBy.Phone`. In custom relationships, the name of the relationship is the value specified in `Field Name` with `__r` appended to it. For example, you can reference contact email from a custom object validation rule using `Contact__r.Email`.

General Tips

- To use a merge field as the destination of a link, insert the merge field after `http://`.
- Salesforce rounds numbers referenced in merge fields according to the user's locale, not the number of decimal spaces specified in the number field configuration.

- You can store the name of an account, contact, or lead in your organization's default language (the local name), in addition to the account or user's default language (the standard name). If the local name is blank, the standard merge field name is used.
- To reference a stand-alone file, use `$Resource.<resource_name>`, where `<resource_name>` is the name you specified when you uploaded the resource.
- If you're using the Translation Workbench to translate custom field names, users can look up merge fields in their chosen language.
- You can't use a lookup field as a merge field in an email template. You can, however, create a hidden formula field on the page layout that pulls the value from the lookup field. Then include the hidden field in the email template.

Merge Fields for Validation Rules

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Syntax and Formatting

When you insert a merge field in a validation rule, the syntax consists of the object, a period, and the field name. For example, `$User.State` corresponds with a user's state or province.

A merge field's syntax can vary depending on where you're using the field. To make sure you're using the correct syntax, select merge fields from the drop-down list in the editor where you're using the merge field. The merge fields for validation rules correspond directly with the fields in your app.

For a list of fields in an object, from the management settings for the object, go to the fields section.

Important:

- If two or more custom objects have matching names or labels, only one of the objects appears when you select from available merge fields. Make sure that all custom objects have unique names and labels so that you can select merge fields from any of the objects.

Limitations

Validation rules can't reference merge fields for:

- Auto number fields, such as `Requisition Number`
- Compound fields, such as addresses, first and last names, dependent picklists, and dependent lookups

  Note: Validation rules can reference merge fields individual address fields, such as `Billing City`.

- Campaign statistic fields, including statistics for individual campaigns and campaign hierarchies

Tips

- Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

  Use the values "Read," "Edit," and "None" in a formula when referencing:

  - $UserRole.CaseAccessForAccountOwner
  - $UserRole.OpportunityAccessForAccountOwner
  - CaseAccessLevel (on Territory)
  - OpportunityAccessLevel (on Territory)

Use the values “Read,” “Edit,” and “All” in a formula when referencing:

- AccountAccessLevel (on Territory)

- Use the `RecordType.Id` merge field in your formula to apply different validations for different record types.

SEE ALSO:

Find Object Management Settings

Merge Fields for Formulas

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Syntax and Formatting

Merge fields for formulas aren’t enclosed in curly braces or preceded by an exclamation point, nor are they preceded by the type of record. For example: `AccountNumber`. To ensure you’re using the correct syntax, use the **Insert Field** button or the drop-down list in the formula editor.

SEE ALSO:

Tips for Using Merge Fields in Formulas

Build a Formula Field

Merge Fields for Cross-Object Formulas

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in all editions

Cross-object formulas are formulas that span two related objects and reference merge fields on those objects. Cross-object formulas can reference merge fields from a master (“parent”) object if an object is on the detail side of a master-detail relationship. Cross-object formulas also work with lookup relationships. For example, you can write a cross-object formula that references the `Account Name` for a contact associated with a case. In this example, you would type `Contact.Account.Name` in a formula on the Case object.

Syntax and Formatting

Merge fields for formulas aren’t enclosed in curly braces or preceded by an exclamation point. Use the relationship names of the objects, not the labels. Although the relationship name is often the same as the object name, it is technically the field name of the relationship field.

To reference the parent account name from Account object, the syntax is `Parent.Name`, not `Account.Name`. When referencing a custom object, add two underscores and the letter *r* to its name. For example, `Position__r.title__c` references the `Job Title` field (`title__c`) on a Position custom object.

Limitations

You can’t reference:

- Merge fields for objects related to activities. For example, merge fields for contacts and accounts are not available in task and event formulas.
- The `$RecordType` global variable—it only resolves to the record containing the formula, not the record to which the formula spans. Starting with the Spring '13 release, when you create a new formula the `$RecordType` global variable is only available for default value formulas.

The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected; however, in list views and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure the formula always returns the intended result. For example:

```
IF
     (OR
          (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.Profile.Name =
"PT2"),
     "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

Automate Your Business Processes

Instead of relying on your users to perform each part of a business process, automate it! The benefits are two-fold: your users can spend their time on other tasks, and you can trust that the process is always done just so. Salesforce offers tools to automate several kinds of business processes: workflow processes, approval processes, wizard-based processes, and more. You'll be happy to know that these tools don't require you to write code—they're all point-and-click.

IN THIS SECTION:

Which Automation Tool Do I Use?

Salesforce provides multiple tools to automate your organization's repetitive business processes: Approvals, Process Builder, Workflow, and Visual Workflow.

The best automation tool for your needs depends on the type of business process that you're automating.

- How a record gets approved

  Example: Managers approve their direct reports' requests for vacation.

- What to do when a record has certain values

  Example: Notify the account owner when a related case is escalated.

- Collecting information from users or customers and then doing something with that information

  Example: Customer support uses a wizard to step through a call script, and cases are created based on the information that they enter.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Processes are available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Flows, approvals, and workflow are available in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How a Record Gets Approved

For example, when an employee requests time off, that time has to be approved by the employee's manager. You need to ensure that when a time-off request is submitted for approval, the right person (the employee's manager) receives the request.

To automate your organization's processes for approving records, create approval processes.

What to Do When a Record Has Certain Values

Three of our tools can address this use case: Workflow, Process Builder, and Visual Workflow. Respectively, these tools create workflow rules, processes, and flows.

We recommend starting with Process Builder, especially for business processes that can be simplified to if/then statements. For example: if a case is escalated, then notify the account owner.

Process Builder includes almost all the functionality that's available in workflow rules, and more. In fact, a single process can do what it would normally take multiple workflow rules to do. The only thing you can do with workflow that you can't do with processes is send outbound messages without code. However, you can work around this limitation by calling Apex code from a process.

If the process is too complicated for the Process Builder or requires more advanced functionality, create a flow by using the Cloud Flow Designer. For example, create a flow to:

- Use complex branching logic (if certain conditions are true, evaluate for further conditions)

  Example: First, check whether a case is escalated. If the case is escalated, check the account's region and route the case accordingly.

- Sort through, iterate over, and operate on several records

  Example: After an opportunity is closed and won, calculate the opportunity's discount. Then apply that discount to all the related opportunity products.

Getting Information from Users or Customers and Then Doing Something with It

If you need to build a wizard to collect information, Visual Workflow is the tool for you. Create a flow that displays information to and requests information from a user. Then take the information that they enter and perform actions in Salesforce with it.

For example, create a flow that walks customer support representatives through a call script. The flow uses information that the representative entered, such as the caller's name and account number, to create a case that's assigned to the right person.

You can add more complexity to the flow to match your business process. For example:

- Route the representative to different screens, depending on earlier choices. This prevents the representative from doing things like trying to upsell a product to a customer who already bought that product.
- Check whether the reported problem is blocking the customer's business and the account is high-value. If so, the flow notifies the region director.

Automation Tool Features

Here's the breakdown of all the features and actions that are supported in each of our automation tools. Use it to figure out which tool is best for your business needs.

	Process Builder	Visual Workflow	Workflow	Approvals
Complexity	Multiple if/then statements	Complex	A single if/then statement	A single if/then statement
Visual designer	✔	✔		
Browser support	All (Chrome recommended)	All (Safari not recommended)	All	All
Starts when	• Record is changed • Invoked by another process	• User clicks button or link • User accesses custom tab • Process starts • Apex is called	Record is changed	• User clicks button or link • Process or flow starts that includes a "Submit for Approval" action • Apex is called
Supports time-based actions	✔	✔	✔	
Supports user interaction		✔		
		Supported Actions		
Call Apex code	✔	✔		
Create records	✔	✔	Tasks only	Tasks only
Invoke processes	✔			
Delete records		✔		
Launch a flow	✔	✔	✔ (Pilot)[1]	
Post to Chatter	✔	✔		

	Process Builder	Visual Workflow	Workflow	Approvals
Send email	✓ (Email alerts only)	✓	✓ (Email alerts only)	✓ (Email alerts only)
Send outbound messages without code			✓	✓
Submit for approval	✓	✓		
Update fields	Any related record	Any record	The record or its parent	The record or its parent

[1]The Process Builder has superseded flow trigger workflow actions, previously available in a pilot program. Orgs that are using flow trigger workflow actions can continue to create and edit them, but they aren't available for new orgs.

Lightning Process Builder

Many of the tasks you assign, the emails you send, and other record updates are vital parts of your organization's standard processes. Instead of doing this repetitive work manually, you can configure processes to do it automatically.

The Process Builder is a workflow tool that helps you easily automate your business processes. Process Builder provides a user-friendly graphical representation of your process as you build it. With the Process Builder, you can:

- Create your processes using a convenient layout with point-and-click efficiency.
- Design an entire process in one place rather than using multiple workflow rules.
- Collaborate with different teams in your organization to create processes.
- Stop using Apex code to automate simple tasks.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Automated processes in the Process Builder are based on records. You can also create processes that invoke other processes. Processes consist of:

- Criteria that determine when to execute action groups
- Immediate actions to execute when criteria are met
- Scheduled actions to execute when criteria are met (non-invocable processes only)

Any change that causes a record or invocable process to match the criteria can automatically trigger the action group. A single process can also execute multiple action groups—so it's easy to automate all of your business records, like accounts, in one place.

Use the more powerful and flexible Process Builder to perform the same actions as workflow. With the Process Builder, you can:

- Create a record
- Update any related record—not just the record or its parent
- Use a quick action to create a record, update a record, or log a call
- Invoke a process from another process
- Launch a flow—you can't schedule this action with workflow

- Send an email
- Post to Chatter
- Submit for approval

If you need your process to do more than what those actions allow, don't worry. You can also call Apex or a flow from a process.

IN THIS SECTION:

Sample Process: Opportunity Management
This example automates a single business process by using the Process Builder instead of workflow rules.

Process Limits and Considerations
Before you start creating, managing, and activating processes, understand the limits and considerations.

Setting Values in the Process Builder
Throughout the Process Builder, you need to set values, for example, to set conditions in a criteria node, to set the fields on a new case in a Create a Record action, or to specify an Apex method to reference.

Setting Advanced Options in the Process Builder
The Process Builder allows you to choose some advanced options for executing actions in your processes.

Create a Process
To create a process, define its properties and which records it should evaluate, and then add criteria nodes and actions.

Troubleshoot Processes
Use the error messages that appear in the Process Builder and the emails you receive when a process fails to help solve problems that arise when you're working with processes. When all else fails, look at the Apex debug logs for your processes.

SEE ALSO:

Which Automation Tool Do I Use?

Sample Process: Opportunity Management

This example automates a single business process by using the Process Builder instead of workflow rules.

The example demonstrates how you can use the Process Builder to automate a single process by adding multiple groups of criteria and then associating individual actions with those criteria. In addition, some actions are available with the Process Builder that you can't perform with workflow rules, such as creating records.

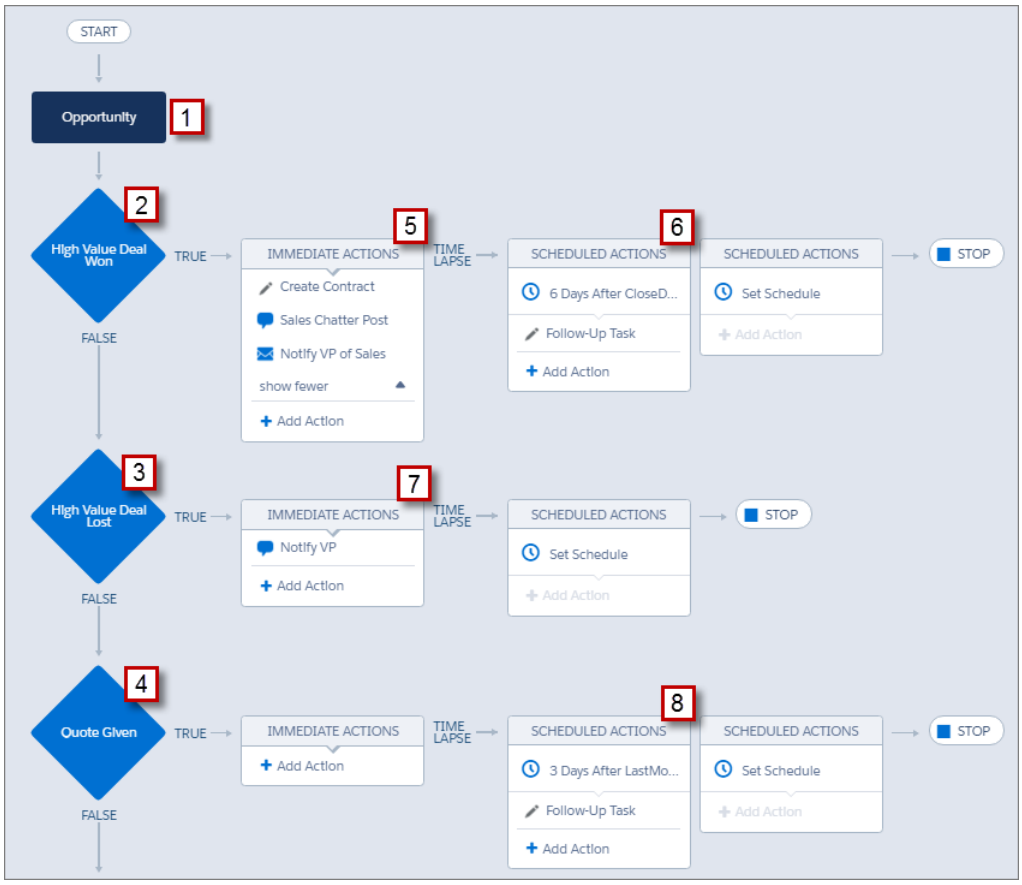
In this example, the process is defined to start when an opportunity record **(1)** is created or edited.

Three criteria nodes are then set up to check whether a high-value deal was won **(2)**, a high-value deal was lost **(3)**, or a quote was given **(4)**. For the first criteria node that evaluates to true, the associated action group is executed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



The High Value Deal Won criteria checks whether the opportunity's stage is closed and won and also whether the opportunity's amount is greater than $1,000,000.00. If both of these conditions are met, the associated action group is executed. For this criteria node, three immediate actions **(5)** and one scheduled action **(6)** are defined.

These actions:

- Create a draft contract record that's associated with the opportunity's account.
- Congratulate the opportunity owner for closing and winning the opportunity by posting to the Sales Chatter group.
- Notify the VP of sales via email that the opportunity was closed and won.
- Create a high priority follow-up task for the associated account's owner, which is scheduled to execute six days after the opportunity's `Close Date`.

If the High Value Deal Won criteria conditions aren't met, the associated group of actions doesn't execute and the next criteria node (High Value Deal Lost) is evaluated.

The High Value Deal Lost criteria node checks whether the opportunity stage is closed and lost and whether the opportunity amount is greater than or equal to $1,000,000.00. If these conditions are true, we've set up an action **(7)** to notify the VP of sales by creating a chatter post on the opportunity record. The post identifies the opportunity and the opportunity amount that was lost.

If neither of the previous criteria conditions are met, the next criteria node defined in this process checks whether the opportunity stage is set to "Proposal/Quote Given." If this condition is true, a scheduled action **(8)** is executed three days after the record is updated. The scheduled action creates a follow-up task for the opportunity owner to make a call to inquire about the opportunity.

Using the Process Builder, we've combined three criteria nodes and associated actions into a single, automated process. To automate the same business process with workflow, you would have to create three different workflow rules and use Apex triggers to create the contract record and post to the Sales Chatter group.

Process Limits and Considerations

Before you start creating, managing, and activating processes, understand the limits and considerations.

Note: We recommend that you use the most recent stable version of Google Chrome™, and your screen resolution must have a width of at least 1024.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Limits for Process Builder

When using the Process Builder, keep shared limits and Apex governor limits in mind.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Limits Shared with Other Features

Processes share some limits with rules and Visual Workflow.

In addition to the following limits, a process's `API Name` must be unique across all processes and flows in your organization.

Description	Per-Organization Limit
Total active rules and active processes per object. Rules include workflow rules, escalation rules, assignment rules, and auto-assignment rules. Invocable processes do not count toward this limit.	50

Description	Per-Organization Limit
Total flows and processes	1,000
Active flows and processes	500
Total number of criteria nodes that are evaluated and actions that are executed at runtime	2,000
Total number of flow interviews that are resumed or groups of scheduled actions that are executed per hour	1000
Total number of flow interviews or groups of scheduled actions that are waiting to be processed	30,000
Total number of relative time alarms defined in flow versions or schedules based on a field value in processes	20,000

The daily limit for emails sent from email alerts is 1,000 per standard Salesforce license per organization—except for Developer Edition organizations, where the daily workflow email limit is 15 per standard Salesforce license. The overall organization limit is 2,000,000. This limit is shared across all features that use workflow email alerts: workflow rules, approval processes, flows, and processes.

Apex Governors and Limits for Processes

Salesforce strictly enforces limits to ensure that runaway processes don't monopolize shared resources in the multitenant environment. Processes are governed by the per-transaction limits that are enforced by Apex. If the process causes the transaction to exceed governor limits, the system rolls back the entire transaction. For details about the operations that are included in the transaction, see "Triggers and Order of Execution" in the *Force.com Apex Developer's Guide*.

Description	Per-Transaction Limit
Total number of SOQL queries issued	100
Total number of records retrieved by SOQL queries	50,000
Total number of DML statements issued	150
Total number of records processed as a result of DML statements	10,000

Each "Create a Record" action uses one DML statement. Each "Quick Action" action uses one DML statement. Each "Update Records" action uses one SOQL query and one DML statement. Each "Flows" action can use multiple SOQL queries and DML statements, depending on the elements that the flow executes. For details, see Limits for Visual Workflow on page 464.

Other Process Limits

Consider these limits when creating and managing processes.

Description	Limit
Total number of characters in a process name	255
Total number of characters in a process's API name	77
Total number of versions of a process	50

Description	Limit
Total number of criteria nodes in a process	200

Process Design Considerations

Before you design a process, understand the limitations and guidelines.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: The Process Builder is supported in Microsoft® Internet Explorer® version 11. Previous versions aren't supported.

- Make sure that your processes aren't set up to create infinite loops. For example, if an Update Records action for Process1 triggers Process2 and a Create a Record action for Process2 triggers Process1, the looping might cause your organization to exceed its hourly limits.
- If you create processes to replace other setup entities, such as a workflow rule, or an Apex trigger, make sure you delete those entities when you activate the equivalent processes. Otherwise, both workflow and processes will fire and cause unexpected results, such as overwritten records or redundant email messages.

  Warning: For a given object, use either workflow rules or processes—not a combination of the two. If you use both, you can't predict the order in which they're executed.
- If you create processes to replace any Apex triggers, make sure you delete those Apex triggers when you activate the equivalent processes. Otherwise, both workflow and processes will fire and cause unexpected results, such as overwritten records or redundant email messages.
- An immediate action on a record update obeys validation rules.
- A scheduled action on a record update skips validation rules.
- Each process is associated with a single object.
- Actions are executed in the order in which they appear in the Process Builder.
- If any of the actions fail, the entire transaction fails and an error message displays. For example, a Post to Chatter action fails if the Chatter group that it tries to post to is private. For details, see Troubleshoot Processes on page 375.
- If a single action group includes multiple "Update Records" actions that apply different values to the same field, the last action's value is used.
- Processes that update owners don't also transfer associated items. To ensure transfer, use one "Update Records" action for each type of child record that you want to transfer. For example, if you're using a process to transfer an account to a new owner, use one action to update all the child contacts, one to update all the child opportunities, one to update all the child contracts, and so on.
- Before you change a custom field's type or name, make sure that it isn't referenced in a process that would be invalidated by the change.

  Note: When an invocable process is associated with an object, you can't change the type for any of the object's fields, even if the process doesn't reference that field.
- If you change a custom field's label, a process that references it won't break. But that process will still show the old label.
- You can't delete a custom field that is referenced by a process.

  Note: When an invocable process is associated with an object, you can't delete any of the object's fields, even if the process doesn't reference that field.
- File type custom fields aren't supported in Process Builder. For example, if your process creates a knowledge article record that has a custom field type of File, an error occurs when the process runs.

- If you have processes on converted leads and want to update the records that result from the conversion, you must enable the lead setting `Require Validation for Converted Leads`.
- If your organization uses multiple currencies, currency fields are updated using the record's currency code. If you choose to update a field based on a formula, any values in your formula are interpreted in the currency code of the record.
- External objects and deprecated custom objects aren't supported in the Process Builder.
- If a standard formula field references a field on a related object, that field's value is always null when a process starts. For example, the `RevenueShare` field on Campaign Influence calculates `CampaignInfluence.Opportunity.Amount * CampaignInfluence.Influence`. Because the formula references a field on Opportunity (a related object), the field's value is null. This limitation isn't the case for custom formula fields that reference a field on a related object. For a custom formula field that uses the same formula, the field's value is derived when a process starts.

Process Chatter Considerations

When creating a Chatter post from a process, consider the limitations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- If your process creates a Chatter post to the feed of a specific user or group, and you deploy the process to another organization, the "Post to Chatter" action may not work because those IDs don't exist in the other organization.
- A Chatter action may fail to save if the `Message` starts with a field reference such as `{![Account].Name}`.

  To work around this issue, add a space at the beginning of the message. For example, if a "Post to Chatter" action's `Message` starts with `{![Account].Name}` in an active process:

  1. Open the active process.
  2. Click **Clone**.
  3. Choose whether to create a version of the current process or a completely new process with its own version history.
  4. Open the "Post to Chatter" action.
  5. In `Message`, enter a space before "`{`".
  6. Save the action.
  7. Save and activate the process. Make sure to deactivate the original process.
- If you use Microsoft® Internet Explorer® version 11, you can't paste text into a message. Copy and paste actions are allowed in all other supported browsers.
- Post to Chatter actions aren't supported if the action posts to a Community user or group. An error occurs when an activated process with this type of action evaluates a record.
- You can add up to 25 mentions to a Chatter message.

Process Scheduled Actions Considerations

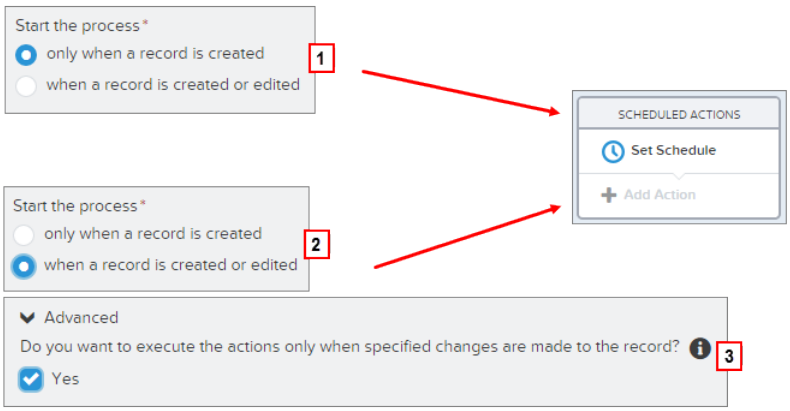
Before you add scheduled actions to a process, understand the limitations and guidelines.

- To add scheduled actions to your process, you have two options:
  - Start the process only when a record is created (**1**). Select this option when you choose an object for your process.
  - Start the process when a record is created or edited (**2**). In addition, select the advanced option to execute actions only when specified changes are made (**3**) when you add criteria to your process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



- A scheduled action on a record update skips validation rules.
- If an action group contains scheduled actions, you can't continue evaluating the next criteria in your process after executing those actions. For details about evaluating additional criteria after executing actions, see Execute Actions on More Than One Criteria.
- If scheduled actions fail to execute—for example, because the user who caused the process to start is inactive—the admin who created the process receives an email with details about the failure. Salesforce makes additional attempts to execute failed scheduled actions before removing them from the queue. For details about error notifications, see Errors Received after a Process Starts Evaluating a Record.
- After you deactivate a process, any scheduled actions continue as usual.
- You can't delete a process if it has unexecuted groups of scheduled actions. For details about deleting unexecuted groups of scheduled actions, see Delete Unexecuted Scheduled Actions on page 374.
- An organization can process up to 1,000 groups of scheduled actions per hour. Each group of scheduled actions is associated with a schedule, such as "3 days from now." When a schedule is processed, the associated actions are executed. If an organization exceeds this limit, Salesforce processes the remaining schedules in the next hour. For example, if an organization has 1,200 groups of pending actions scheduled to be processed between 4:00 PM and 5:00 PM, Salesforce processes 1000 groups between 4:00 PM and 5:00 PM and the additional 200 groups between 5:00 PM and 6:00 PM.
- An organization can have up to 30,000 pending schedules and waiting flow interviews at one time.
- For processes that are set to run when a record is created or edited, scheduled actions remain in the queue only as long as the criteria for those actions are still valid. If a record no longer matches the criteria, Salesforce removes the scheduled actions for that record from the queue.
- For processes that are set to run when a record is created, Salesforce never reevaluates the record with that process. Scheduled actions remain in the queue, even if the record no longer meets the associated criteria when the scheduled actions are executed.

- If you schedule an action for *0* **Days Before** a date, save the process, and then reopen the process, the schedule in the process changes to *0* **Days After** the date. The process still executes at the specified time.
- If you schedule an action for *0* **Days After** a date, there may be a delay from the time represented by the date field before the action group executes.

Transactions and Scheduled Actions

Similar to workflow rules, immediate actions in processes are executed in the same transaction as the operation that caused the process to be triggered—when a user created or edited a record. For details about the operations that are included in that transaction, see Triggers and Order of Execution in the *Force.com Apex Developer's Guide*. Scheduled actions are included in a separate transaction.

Groups of scheduled actions aren't performed independently. They're grouped into a single batch that starts performing within one hour after the first group enters the batch. The batch can also include flow interviews that are resumed after a specified time occurs. This behavior can cause you to exceed your Apex governor limits if any actions in the group execute DML operations or SOQL queries. A DML operation is used each time a Salesforce record is created, updated, or deleted, such as when a process executes a "Create a Record" action. A SOQL query is used each time Salesforce looks up information about an existing record, such as when a process executes an "Update Records" action. For details on Apex governor limits, see Limits for Process Builder on page 336.

If your process contains scheduled actions, make sure that the actions don't perform more DML operations or SOQL queries than are permitted by the Apex governor limits.

If a group of scheduled actions fails to be executed:

- Prior groups of scheduled actions in that batch's transaction are successful.
- The immediate actions for that process are successful.
- All the scheduled actions in that group fail to be executed.
- The remaining groups of scheduled actions in that batch are tried.

Example: Salesforce is going to process a batch of 25 groups of scheduled actions in the same transaction. Up to 100 DML operations can be used across this batch. The first 22 groups each use five (5) DML operations, and the last three (3) groups don't use any DML operations.

The 21st group violates the Apex governor limits by trying to execute the 101st through 105th DML operations for this transaction. Only this group of scheduled actions fails to execute. This means that the first 20 groups of scheduled actions are successfully executed and the 21st group of scheduled actions fails. None of the actions in the 21st group is successful, no matter which action in the group violated the limit. Salesforce then tries to execute the remaining groups of scheduled actions in the batch. Because the 22nd group uses five (5) DML operations and the transaction has already used all of its allowed DML operations, the 22nd group also fails. Because the last three (3) groups don't use any DML operations, those groups are successfully executed.

Schedule Limitations

- If actions are scheduled for a time in the past, Salesforce executes that group of scheduled actions within one hour.

  For example, if a process is configured to email an opportunity owner seven days before the close date and the process runs for an opportunity with the close date set to today, Salesforce executes the scheduled actions within an hour.

If actions are scheduled based on the current time (i.e. 3 days from now):

- The schedule is evaluated based on the timezone of the user who created the process.

If actions are scheduled based on a field value (i.e. 3 days after a case's `Created Date`):

- The schedule is evaluated based on the organization's timezone.
- If a deactivated process still has pending scheduled actions and the record whose field the schedule is based on is changed, Salesforce recalculates the schedule for those actions. Once a process is deactivated, Salesforce ignores all other changes to associated records.

- Across all of your processes and flow versions, your organization can have up to 20,000 schedules based on a field and relative time alarms.
- The referenced field can't be a:
  - DATE or DATETIME field that contains automatically derived functions, such as `TODAY` or `NOW`.
  - Formula field that includes related-object merge fields.
- If you change the referenced field value and the schedule hasn't been processed, Salesforce recalculates the schedule associated with that field.

  For example, if a process is configured to email an opportunity owner seven days before the opportunity close date and the close date is set to 2/20/2014, Salesforce processes the schedule on 2/13/2014 and sends the email. If the close date is later updated to 2/10/2014 and the schedule hasn't been processed yet, Salesforce recalculates the schedule and sends the email on 2/3/2014. If Salesforce recalculates the schedule to a date in the past, Salesforce executes the associated actions shortly after you save the record.
- Salesforce ignores schedules that reference null field values.
- If the record or the object that the schedule is associated with is deleted, the schedule is never processed.
- The following limitations apply for converted leads.
  - You can't convert a lead if an unexecuted schedule is based on one of the lead's fields.
  - If `Validation and Triggers from Lead Convert` is enabled, scheduled actions on leads aren't executed during lead conversion.
  - If a campaign member based on a lead is converted before scheduled actions that are associated with that record finish, Salesforce still executes the scheduled actions.
- The SignupRequest object isn't supported in scheduled actions.

Process Builder Accessibility Considerations

Process Builder is 508-compliant, with one exception.

You can close modal dialogs using the ESC key on your keyboard, but you can't close side panels by using the ESC key.

Tip: You can change the order of criteria nodes in your process.

- To select criteria nodes, press Space.
- To change the order of a criteria node, use the up and down arrows.
- To save your changes, press Space.
- To cancel, press ESC.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Change Sets and Packaging Considerations for Processes

Keep these considerations in mind when deploying processes.

- The Metadata API allows you to deploy processes to another organization. However, keep in mind that:
  - If the process already exists in the target organization, it must be inactive or it won't deploy.
  - If the process doesn't exist in the target organization, it will deploy with an inactive status regardless of its status in the source organization.
- If a process is deployed to another organization in a managed package, you can only activate it or deactivate it. You can't view or edit the process.
- If you deploy a process through the Metadata API or change sets and the process contains any of the following actions, packageable components that they reference aren't included in the package or change set automatically. To deploy successfully, manually add those referenced components to the package or change set.
  - Apex
  - Email Alerts
  - Launch a Flow
  - Post to Chatter
  - Quick Actions
  - Submit for Approval

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Process Formula Limitations

Formulas that are used as conditions in a criteria node have some limitations. If a process contains an invalid formula, you can't save or activate the process.

All formulas that are used in a criteria node must:

- Return `true` or `false`. If the formula returns `true`, the associated actions are executed.
- Not contain more than 3,000 characters.
- Not contain an unsupported function.

Tip: Parentheses aren't included automatically when you insert a function. Be sure to add parentheses, for example, *`TODAY()`*, when building a formula.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Unsupported Functions

If a formula in a process uses any of the following functions, the formula will return `null`.

- GETRECORDIDS
- IMAGE
- INCLUDE
- PARENTGROUPVAL
- PREVGROUPVAL
- REQUIRE SCRIPT
- VLOOKUP

For a complete list of operators and functions for building formulas in Salesforce, see Formula Operators and Functions on page 234.

Note:

- If your process criteria uses a formula, don't create a formula that always evaluates to true, such as `2 < 5`.
- ISCHANGED is available as both a formula function and as an operator. When it's used as a formula function in process criteria, you can't reference a child record's related fields. For example, ISCHANGED isn't supported when referencing a `[Case].Contact.AccountId` field, but it can be used when referencing `[Case].ContactId`.

SEE ALSO:

Tips for Working with Picklist and Multi-Select Picklist Formula Fields

Process Builder Advanced Option Considerations

Tips for Working with Picklist and Multi-Select Picklist Formula Fields

Process Activation Considerations

Before you activate a process, understand when processes start evaluating records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- After you activate a process, you can no longer edit that process.
- Salesforce processes rules in the following order:
  1. Validation rules
  2. Assignment rules
  3. Auto-response rules
  4. Workflow rules and processes (and their immediate actions)
  5. Escalation rules
- Processes start automatically and are invisible to the user.
- Saving or creating records can trigger more than one process.

When Do Processes Start Evaluating Records?

- Processes can evaluate a record anytime a record is saved or created. However, processes that are created after records are saved don't evaluate those records retroactively.
- Processes can evaluate records up to five times in a single transaction if another process, workflow rule, or flow updates the record in the same transaction. See Reevaluate Records in the Process Builder on page 350 for more details.
- Processes are triggered when a standard object in a master-detail relationship is re-parented.
- Processes only evaluate converted leads if validation and triggers for lead convert are enabled in your organization.
- Processes evaluate changes made to records while using Connect Offline when users synchronize.
- In a batch update, processes are only retriggered on the entities where there is a change.
- The following actions don't trigger processes:
  - Mass replacing picklist values
  - Mass updating address fields
  - Mass updating divisions
  - Changing the territory assignments of accounts and opportunities
  - Deactivating Self-Service Portal, Customer Portal, or partner portal users
  - Converting state and country data by using the Convert tool
  - Making changes to state and country picklists using `AddressSettings` in the Metadata API

Process Builder Error Messages

Error or warning messages might refer to a "flow" instead of a "process." Those messages still apply to your process.

Keep these considerations in mind when errors occur.

- If you create or edit a person account and your process uses an account field value (for example, `Account.Name Equals Acme`), users might see this error. "Workflow Action Failed to Trigger Flow The record couldn't be saved because it failed to trigger a flow." You can deactivate your process to avoid this error.
- A reference field contains an ID value that points to a unique record on another object. Errors occur if that ID value isn't set. For example:
  - Your process has a "Post to Chatter" action with a message that references an account name related to a contact record. `A new contact for account {![Contact].Account.Name} was created.` If the ID value for the account reference field on the contact record isn't set, an error occurs when the process runs.
  - Your process criteria checks whether the `Name` field on an account related to a contact record equals "Acme" (`[Contact].Account.Name Equals Acme`). If a user creates a contact record (related to an account) without setting the ID value, an error occurs. In this case, you can avoid the error by adding an additional filter condition above `[Contact].Account.Name Equals Acme` as follows: `[Contact].AccountId Is null False`.
- If your process criteria evaluates a lookup relationship field and that field value is empty, the process may fail.

  For example, your process evaluates a custom lookup relationship field that references the status of a custom bug record related to a case. If the bug's status field is empty, the process fails and stops evaluating any remaining fields in the order listed in your process criteria. The process correctly evaluates the same lookup relationship field if you use an equivalent formula in your process criteria. If your process criteria doesn't use a formula, we recommend checking for isNull when referencing fields that traverse multiple related records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Troubleshoot Processes

Identifying Errors in the Process Builder

Errors Received after a Process Starts Evaluating a Record

Setting Values in the Process Builder

Throughout the Process Builder, you need to set values, for example, to set conditions in a criteria node, to set the fields on a new case in a Create a Record action, or to specify an Apex method to reference.

IN THIS SECTION:

Field Picker

Use the field picker to reference fields on the record that started the process or fields on related records.

Process Builder Value Types

When setting a value for a given field—whether on the record that started the process or a related record— the available value types are filtered based on the field that you've selected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

[Multi-Select Picklists in the Process Builder]
The Process Builder lets you select multiple values for a multi-select picklist field.

Field Picker

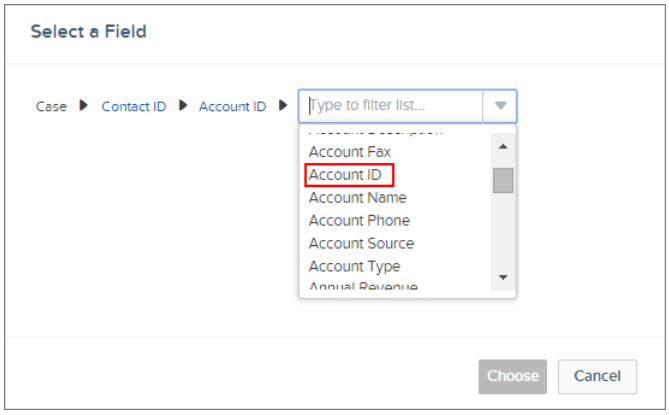
Use the field picker to reference fields on the record that started the process or fields on related records.

To use fields on a related record, click a field with ❯ next to the value. For example, use the Account ID field value on the case's contact's account.

EDITIONS

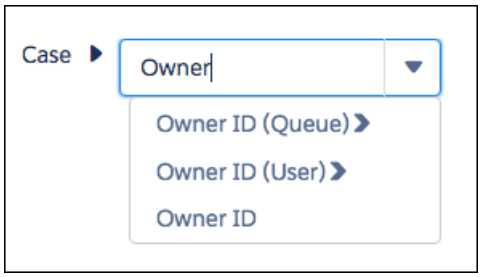
Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



The field picker displays only fields that are compatible with the selected parameter.

If you see a field multiple times, that means the field can relate to multiple objects. For example, if you created a queue for cases, a case's owner can be either a user or a queue. `Owner ID` is a *polymorphic field*—a field that relates to more than one object.



To access a field on the case's owner, choose the appropriate relationship. Be careful, though. If you pick **Owner ID (User)** and the owner of the record is actually a queue, the process fails.

Process Builder Value Types

When setting a value for a given field—whether on the record that started the process or a related record— the available value types are filtered based on the field that you've selected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The available value types are:

- **Currency**—manually enter a currency value.
- **Boolean**—choose a true/false boolean value.
- **DateTime** or **Date**—manually enter a date/time or date value.
- **Formula**—create a formula expression.
- **Global Constant**—choose a global constant to set a value to null or an empty string—for example, choose $GlobalConstant.Null.
- **ID**—manually enter a Salesforce ID value, for example, *00300000003T2PGAA0*.

  Note: If your process is based on a user ID (for example, when an [Event].OwnerId equals a specific ID value) make sure that the ID value is an 18-character ID and not a 15-character ID. You can convert a 15-character ID to 18 characters at www.adminbooster.com/tool/15to18.

- **MultiPicklist**—choose one or more multi-select picklist values.
- **Number**—manually enter a number value.
- **Picklist**—choose a picklist value.
- **Reference**—choose a field on the record or on a related record.
- **String**—manually enter a string value.

Note: These global constant values aren't supported with the "is null" operator.

- *$GlobalConstant.Null*
- *$GlobalConstant.EmptyString*

Multi-Select Picklists in the Process Builder

The Process Builder lets you select multiple values for a multi-select picklist field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

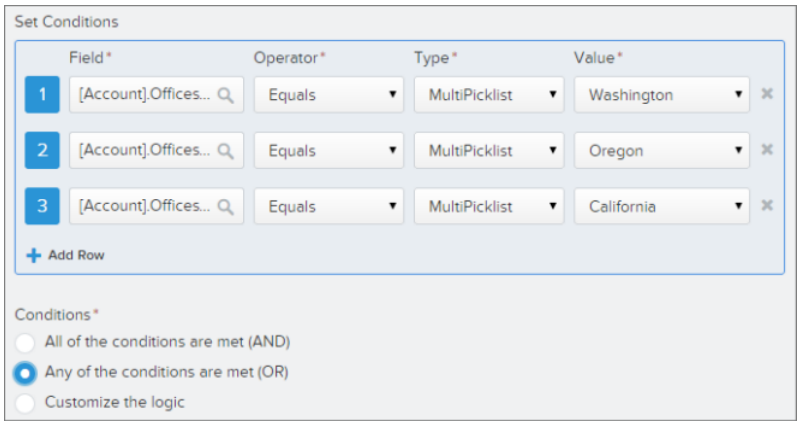
Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

For example, set multiple values for the `Country` field for a company record that operates in Ireland, the UK, and France.

You can use multi-select picklists in:

- Formulas
- Process criteria
- Create a Record actions
- Quick Actions
- Update Records actions

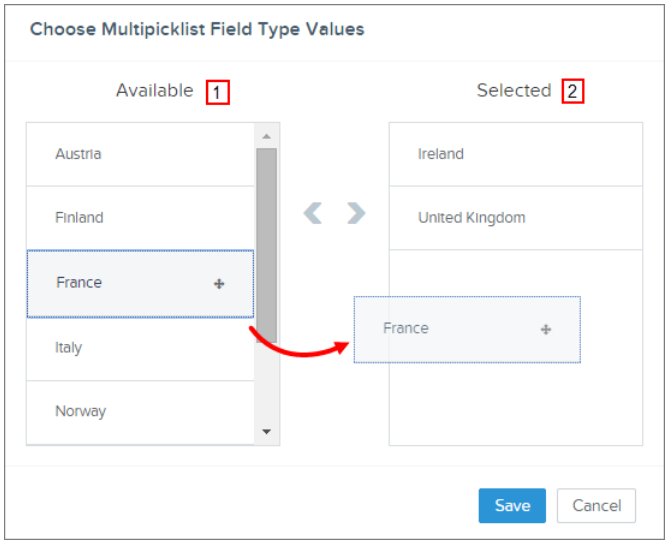
In process criteria, set multiple values by creating one condition for each individual multi-select picklist value. For example, if your process checks whether changes were made to an account's office locations, reference multiple values by choosing the same field for each multi-select picklist value.



Refer to Tips for Working with Picklist and Multi-Select Picklist Formula Fields for more information.

When you reference a multi-select picklist field in an action, enter values by clicking **Choose values…**

Add or remove values by dragging them between the Available (**1**) and Selected (**2**) columns.



Keep these considerations in mind when using operators with multi-select picklists.

- You can use the Equals operator if you select only one value from a multi-select picklist field.
- If you use the Equals operator with multiple multi-select picklist values and choose the **Any of the conditions are met (OR)** option, the condition matches on one value only. For example, if your process checks whether a Region field equals West or East, the condition evaluates to true when the value is West or when the value is East, but won't evaluate to true when both West and East are selected values.
- If you use **Contains** and **OR** to evaluate multiple multi-select picklist values, the condition evaluates to true on multiple values. For example, if your process checks whether a `Region` field contains West or East, the condition evaluates to true when a `Region` field contains West and East or when a `Region` field contains West or East values.

Setting Advanced Options in the Process Builder

The Process Builder allows you to choose some advanced options for executing actions in your processes.

Choose these options by expanding the **Advanced** area of the side panel when you:

- Add objects to your noninvocable process
- Add criteria to your process

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Process Builder Advanced Option Considerations
Keep these considerations in mind when choosing advanced options.

Reevaluate Records in the Process Builder
When you add objects to your process, you can choose to evaluate a record multiple times in a single transaction.

Nest Processes in the Process Builder
Invocable processes let you modularize sections of your processes and add more logic to them. An *invocable process* is process that starts when another process invokes it. Whether a process is invocable or not is controlled by `The process starts when` in the process's properties.

Avoid Unwanted Actions in Processes
When you add criteria to your process, you can choose to execute actions only when specified criteria changes.

Process Builder Advanced Option Considerations

Keep these considerations in mind when choosing advanced options.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Avoid creating an infinite loop when allowing your process to reevaluate records. For example, if your process checks whether an account description changes and then updates an account description and creates a Chatter post every time an account record is created or edited, the process will evaluate and trigger actions resulting in six Chatter posts.
- If you choose to evaluate a record multiple times in a single transaction when you specify an object for your process, we recommend not setting any of your criteria to No criteria—just execute the actions!.
- We recommend that you don't use the advanced options if your process uses ISCHANGED, ISNEW, or PRIORVALUE formula functions. If you do use these options, keep these considerations in mind.
  - When a record is first created, ISNEW evaluates to true throughout a process transaction. If your process uses the ISNEW formula function and reevaluates a record multiple times in a single transaction, the process may execute actions multiple times.

    For example, your process checks whether an account is created or updated.

    1. When ISNEW evaluates to true, the process updates the account's annual revenue and posts to Chatter.
    2. When the process updates the account's annual revenue, the process then reevaluates the record (up to five additional times) because the record was changed.

    Each time the criteria is evaluated, ISNEW evaluates to true. The results is six Chatter posts. This example is true only if the process is triggered because an account record is created.
  - When a record is created, PRIORVALUE returns the current value as the prior value. When a record is updated, PRIORVALUE returns the field value that was set immediately before the transaction started. If your process uses the PRIORVALUE formula

function and reevaluates a record multiple times in a single transaction, the process may execute actions multiple times. If your process reevaluates a record multiple times in a single transaction and executes actions only when specified criteria changes, the prior value returns the values that existed before the record was saved.

For example, your process checks whether an account is created or updated.

1. When `PRIORVALUE([Account].Type) = 'Prospect'` evaluates to true, the process updates the account's annual revenue and posts to Chatter.
2. When an account is created with `Prospect` as the account type, the criteria is always true until the end of the process transaction.
3. If the process is changed to update the account type to `Other` when the criteria is true, then for an account created with `Prospect` as the account type, the formula `PRIORVALUE([Account].Type) = 'Prospect'` will be true until the end of process transaction.

Each time the record is reevaluated, the prior value of the account's type is Prospect. The result is six Chatter posts.

- ISCHANGED always evaluates to false when a record is first created.

  For example, your process checks whether an account description changes—`ISCHANGED([Account].Description)`—and the process also reevaluates records multiple times in a single transaction. If an account is first created with a blank description value and another process updates the account description in the same transaction, ISCHANGED evaluates to true every time the record is reevaluated because it compares the account description value when the record was first created (a blank value) with whatever's set for the current value.

  Let's say this same process creates a Chatter post every time ISCHANGED([Account].Description) evaluates to true. This process would create a recursive loop resulting in six Chatter posts because ISCHANGED evaluates to true throughout the transaction.

Reevaluate Records in the Process Builder

When you add objects to your process, you can choose to evaluate a record multiple times in a single transaction.

It's kind of like using a roundabout instead of a four-way stop to control process traffic. Instead of stopping and waiting for separate transactions, reevaluating records helps your business traffic flow a little more freely.

If you choose this option, the process can evaluate the same record up to five additional times in a single transaction. It might reevaluate the record because a process, workflow rule, or flow updated the record in the same transaction. When a record is reevaluated, the process uses the most recent values for that record.

For example, your sales review process includes multiple steps, approvals, notifications, and fields that need to be updated. Some of these changes may be part of your process, or they may be managed by other workflow rules or flows in your organization. If you allow the Process Builder to reevaluate a record multiple times in a single transaction, you can manage and evaluate all of these changes—even changes from other processes—in a single transaction in your process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Process Builder Advanced Option Considerations

Nest Processes in the Process Builder

Invocable processes let you modularize sections of your processes and add more logic to them. An *invocable process* is process that starts when another process invokes it. Whether a process is invocable or not is controlled by `The process starts when` in the process's properties.

To invoke a process from another process, you configure a "Processes" action. That configuration includes passing a record to the invocable process. That's how the process knows which record to start with. Because the record is passed from one process to another, the invocable process receives a certain version of that record. That version differs, depending on when the "Processes" action is executed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Immediate Action

When you invoke a process through an immediate action, the process receives the values that the record contained when the top-level process starts.

Example: Process 1 updates an account and then invokes Process 2 based on that account. Process 2 receives the version of the account when Process 1 started.

Scheduled Action

When you invoke a process through a scheduled action, the process receives the latest values for the record.

Example: Process 1 updates an account and, 15 minutes later, invokes Process 2 based on that account. Process 2 receives the latest version of the account from the database.

When Should I Build an Invocable Process?

Do you find yourself building the exact same actions for multiple action groups? Configure those actions one time in an invocable process, and then invoke that process from all the relevant action groups. Later, to update those actions, update the one invocable process. All the other processes then use the updated actions.

Another cool scenario for invocable processes: nesting simple logic. Processes handle simple "if/then" statements. But what if you need to nest some of those statements? Rather than having to build a flow or write code, build the second level of logic into another process. Invoke the second process from the first, and voila!

Example: Let's say you handle all of your case management in a single process. But you need to treat escalated cases for high-revenue accounts differently from escalated cases for regular accounts. If the case is escalated by an account whose renewal date is less than a month away, notify the account owner, the regional manager, and the VP of that region. If the case is escalated by an account whose renewal date is more than a month away, notify only the account owner and the regional manager.

To do so, you build an invocable process. Let's call it "Escalated Cases." The process operates on the Case object and has two criteria nodes.

- The first criteria node evaluates whether the associated account's renewal date is less than a month away. When a case meets that criteria, the process posts to the account's feed with a link to the case and mentions the account owner, regional manager, and regional VP.
- The second criteria node has no criteria. If a case doesn't meet the first node's criteria, the process performs the same action, except that it doesn't mention the regional VP.

Now back to the process that automates your case management. You already have a criteria node that checks whether the case is escalated. Add a "Processes" action to that criteria's action group, and configure the action to invoke the "Escalated Cases" process.

Avoid Unwanted Actions in Processes

When you add criteria to your process, you can choose to execute actions only when specified criteria changes.

For example, your process sends an email alert whenever a case has an Escalated status. Let's say your support team repeatedly updates the case description with new information. Whenever the case is saved with a new description, you process can check specifically whether the Escalated status changed instead of just repeatedly sending email alerts. This way, the process executes actions only if the status wasn't set to Escalated and was just changed to Escalated.

Tip: Check out this short video Avoid Unwanted Actions in Your Process to learn more about this option.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

This setting isn't supported if:

- Your process starts only when a record is created.
- Your process starts when a record is created or edited and the criteria node doesn't evaluate any criteria.
- The criteria node evaluates a formula, but the formula doesn't include a reference to the record that started the process.
- Your process uses the Is changed operator in a filter condition.

If Yes is...	Actions are executed if...	Actions are not executed if...
Selected	• The record was created. • The record was updated. Its current values meet the conditions, and its most recent previous values did not meet the conditions.	• The record's current values meet the conditions, and the record's most recent previous values met the criteria. • The record's current values don't meet the conditions.
Deselected	• The record was created. • The record was updated, and its current values meet the conditions.	The record's current values don't meet the conditions.

SEE ALSO:

Process Builder Advanced Option Considerations

Create a Process

To create a process, define its properties and which records it should evaluate, and then add criteria nodes and actions.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

IN THIS SECTION:

1. Define the Process Properties
   Define the process properties to uniquely identify and help you manage your processes.
2. Identify Which Records to Evaluate
   Associate the process with an object, and specify when to start the process.
3. Add Process Criteria
   Define the criteria that must be true before the process can execute the associated actions.
4. Add Actions to Your Process
   After you define a criteria node, define the actions that are executed when the criteria are met.
5. Execute Actions on More Than One Criteria
   Choose whether to stop or continue your process after specific criteria are met and associated actions execute.

Define the Process Properties

Define the process properties to uniquely identify and help you manage your processes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

1. From Setup, enter *Process Builder* in the Quick Find box, select **Process Builder**, and then click **New**.

   You can also modify an existing active process by cloning a new inactive copy of it. The copy can be a new process or a new version of the current process.

2. Fill out these fields to define your process.

Field	Description
Process Name	The name for your process. Process Name must be 255 characters or fewer. This name appears in the process management page (from Setup, this page is located by entering *Process Builder* in the Quick Find box, then selecting **Process Builder**), so consider naming your process so that you can differentiate it from other processes.
API Name	The name that's used by the API and managed packages. API Name must be 79 characters or fewer. This name must be unique across all processes and flows. (In flows, this field is Unique Name.) The name must begin with a letter and use only alphanumeric characters and underscores. It can't include spaces, end with an underscore, or have two consecutive underscores. After it's saved, API Name can't be changed for the process.

Field	Description
`Description`	Optional. A description for your process. The description also appears in the process management page. It's intended to help you differentiate between processes, such as to understand what a process does.
`The process starts when`	Identifies when the process begins. You can set your process to start when: • A record changes • It's invoked by another process

3. Click **Save**.

Identify Which Records to Evaluate

Associate the process with an object, and specify when to start the process.

1. Click **Add Object**.
2. For `Object`, choose the object that you want to base this process on. Click **Find an object...** to open the drop-down list and type to filter the list.
   After you save this panel, you can't change the selected object.
3. Select when you need the process to start.

If you want the process to start every time that...	Select
A record of the specified object type is created but not updated.	only when a record is created
A record of the specified object type is created or updated.	when a record is created or edited

Note: If you're familiar with workflow rules and you're wondering what happened to the third option ("created, and any time it's edited to subsequently meet criteria"), don't worry! You'll get to that setting when you add criteria nodes if you select "when a record is created or edited" here.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:
- "Manage Force.com Flow"

  AND

  "View All Data"

4. To choose whether you want the process to evaluate a record multiple times in a single transaction, expand the **Advanced** area.
   If you select yes, the process can evaluate the same record up to five additional times in a single transaction. It might reevaluate the record because a process, workflow rule, or flow updated the record in the same transaction. See Reevaluate Records in the Process Builder on page 350 for more information.
5. Click **Save**.

Add Process Criteria

Define the criteria that must be true before the process can execute the associated actions.

When criteria are met, the process executes the associated action group. When criteria aren't met, the process skips the action group and evaluates the criteria for the next action group.

1. Click **Add Criteria**.
2. Enter a name for this criteria node.

   The name appears on the canvas, so use a name that helps you differentiate between other criteria nodes.
3. Select the type of criteria that you need to define. This selection determines which fields appear later in the dialog box.

If you need...	Select
The record to have certain field values. For example, to execute the associated actions on opportunity records with an amount greater than $5,000, set the filter to: `[Opportunity].Amount greater than $5000.00`	Conditions are met
To evaluate the record by using a formula. For example, to execute the associated actions on accounts whose annual revenue is over $1,000,000 when the account is changed by someone other than the owner, use this formula. `AND (([Account].LastModifiedBy.Id <> [Account].Owner.Id) , ([Account].AnnualRevenue > 1000000) )`	Formula evaluates to true
To simply execute the associated actions without evaluating the record. The process executes all the actions that are associated with this criteria node and, unless you specify otherwise, doesn't evaluate any remaining criteria nodes in the process. If you choose to stop your process after executing these actions, we recommend choosing this option for only the last criteria node in your process.	No criteria—just execute the actions!

4. If you selected "Conditions are met":
   a. Define the filter conditions by identifying the field values that the process needs to evaluate.

`Field`	Select the field whose value you want to evaluate. You can also evaluate values for records that are related to the one that started the process. To do so, click on a related record with › next to the ID field. For example, if a contact record started the process, you can evaluate the value for the contact's account's `Annual Revenue` field. To access that field, click `Account Id` ›, select **Annual Revenue**, and then click **Choose**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

`Operator`	The available operators depend on the field's data type.
`Type`	The available value types depend on the field' data type.
`Value`	Identify the value that you want to evaluate the field for. See Setting Values in the Process Builder on page 345 for details.

b. For `Conditions`, identify which conditions must be true for the process to execute the associated actions.

If you choose to use custom logic, enter a text string using:

- Numbers to refer to each condition
- AND or OR to identify whether all or just one of the conditions must true
- Parentheses to group parts of the string together

For example, if you enter 1 AND (2 OR 3), the outcome evaluates to true if the first condition is true and either the second or third outcome is true. The maximum length of the condition logic text is 1000 characters as per Flow restrictions.

Note: Ambiguous logic separated by a parentheses may not cause a validation error. For example:

- 1 AND 2 OR 3 results in an error
- 1 AND (2 AND 3) OR 4 doesn't result in an error

Avoid creating ambiguous custom logic when grouping parts of a string or you may see unexpected results.

5. If you selected "Formula evaluates to true," define the formula.
6. Optionally, to specify whether you want to execute the actions only if the record was created or edited to meet criteria, click **Advanced** at the bottom of the panel .

   For details, see Avoid Unwanted Actions in Processes on page 352.

   Note: This setting is available only if the process starts when a record is created or edited and you selected "Filter conditions are met" or "Formula evaluates to true."
7. Click **Save**.

SEE ALSO:

Execute Actions on More Than One Criteria

Add Actions to Your Process

After you define a criteria node, define the actions that are executed when the criteria are met.

An action group can consist of a combination of immediate and scheduled actions. Immediate actions are executed when evaluation criteria are met. Scheduled actions are executed at a specified time. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open 10 days before the specified close date.

Before you begin, consider whether you want this action to be executed immediately or at a specific time. If you want to execute the action at a specific time, identify when those actions should be executed.

1. Click **Add Action**.
2. Select the type of action to create, and then fill out the fields to define the action.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

IN THIS SECTION:

Create a Record from a Process

Create a record by manually entering values or by using the values of related records.

Invoke a Process from Another Process

Invoke a process from another process. With invocable processes, you have the option of reuse so that you don't spend your time on repetitive work.

Create a Chatter Post from a Process

Post to the feed of a user, a Chatter group, or the record that started the process.

Use a Quick Action from a Process

Create a record, update a record, or log a call by using a quick action that you or another administrator created for your organization.

Launch a Flow from a Process

Start an autolaunched flow from your process to automate complex business processes—create flows to perform logic and have events trigger the flows via processes—without writing code.

Send an Email from a Process

Easily send an email from a process by using an email alert. Email alerts are configured outside of the Process Builder and contain the standard text, list of recipients, and template for an email.

Submit a Record for Approval from a Process

Submit the record that started the process for approval.

Update Records from a Process

Update one or more records that are related to the record that started the process by manually entering values or by using the values from related records.

Call Apex Code from a Process

Add customized functionality to your process by calling Apex from the process.

Specify When Your Actions Should Execute

If you want some actions to be executed at a specific time, first define the schedule for those actions.

Note: Scheduled actions are complex features. See Scheduled Actions Considerations for more details.

Each action group that supports scheduled actions can have multiple schedules. For example, you can schedule some actions to be executed one day from now and other actions to be executed three days from now.

To identify when an action group's scheduled actions should be executed:

1. In an action group that supports scheduled actions, click **Set Schedule**.
2. If you need to schedule actions based on a date/time field on the record that started the process:
   a. Leave the first radio button selected.
   b. Click the drop-down list on the right side of the panel, and select the date that you want to use to schedule the action.
   For example, if your process is based on an account record, choose the account's `Created Date`.
   c. Specify a number of days or hours before or after the field.
   If the criteria for this action group are still met when this time occurs, Salesforce executes the scheduled actions.
3. If you need to schedule actions to be executed a certain number of days or hours from when the process is executed:
   a. Select the second radio button.
   b. Specify the number of days or hours from when the process is executed.
   If the criteria for this action group are still met when this time occurs, Salesforce executes the scheduled actions.
4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"
  AND
  "View All Data"

Create a Record from a Process

Create a record by manually entering values or by using the values of related records.

After you create an action and select "Create a Record" for the type, fill in the relevant fields to add the action to your process. The new record's `Created By` field is then set to the user who started the process by creating or editing a record.

Warning: If you create processes to replace any workflow rules, ensure that you delete those workflow rules when you activate the equivalent processes. Otherwise, both workflow rules and processes fire and cause unexpected results, such as overwritten records or redundant email messages. Do the same if you create processes to replace any Apex triggers.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. For `Record Type`, select the object that you want to create a record for. To filter the drop-down list, type the name of the object to filter the drop-down list.

   When you select an object, at least one row appears to allow you to set field values for the new record.

   Warning: Rows appear automatically for fields required by the API. You might need to provide values for other fields. We recommend that you refer to your organization's page layouts to determine which fields are required.

3. Set the record's field values.

`Field`	Select the field whose value you want to set. To filter the drop-down list, type the name of the field.
`Type`	Select the type of value that you want to use. The available types depend on the field that you've selected.
`Value`	Set a value for the field. by using the text entry field to manually enter a value or the field picker to use a field value from a related record. See Setting Values in the Process Builder on page 345 for details.

4. Click **Save**.

Tip:

- If you set up your process to create an account record, *`Name`* appears as a required field. If you want to create a person account, you can add *`LastName`* as a field but it doesn't appear as required by default. You can enter a dummy value for the *`Name`* field.
- When you create a record, required fields normally appear at the top of the list. However, if you save a Create a Record action, close the process, and then reopen the action, required fields may not appear in the normal order.

Invoke a Process from Another Process

Invoke a process from another process. With invocable processes, you have the option of reuse so that you don't spend your time on repetitive work.

After you create an action and select "Processes" for the type, fill in the relevant fields to add the action to your process.

You can invoke processes with objects that share at least one unique ID. For example, in the Account and Case objects, the `AccountId` field is unique to Account and also used by Case. You can create an invocable process that updates a Case record. Then you can invoke it from:

- A process that updates an Account record's owner
- A process that adds an Account shipping address or updates it

When you create a process that invokes another process, each one counts toward your process and other applicable limits. DML limits in processes that invoke processes count as one transaction.

Warning: If you create processes to replace any workflow rules, delete those workflow rules when you activate the equivalent processes. Otherwise, both workflow rules and processes fire and cause unexpected results, such as overwritten records or redundant email messages. Do the same if you create processes to replace any Apex triggers.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. Select an invocable process. You can only select active invocable processes.
3. Select your process variable. Remember that you can only select fields related to the object associated with the process you invoke.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Create a Chatter Post from a Process

Post to the feed of a user, a Chatter group, or the record that started the process.

This action is available only if your organization has Chatter enabled.

IN THIS SECTION:

Post to a User's Chatter Feed from a Process
Post to the feed of a user by identifying a specific user in your organization or a User lookup field on a record.

Post to a Chatter Group from a Process
Post to the feed of a specific Chatter group.

Post to a Record's Chatter Feed from a Process
Post to the feed of the record that started the process.

SEE ALSO:

Process Chatter Considerations

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Post to a User's Chatter Feed from a Process

Post to the feed of a user by identifying a specific user in your organization or a User lookup field on a record.

After you've created an action and selected "Post to Chatter" for the action type, fill in the relevant fields to add the action to your process. The feed item will appear as if the user who started the process—by creating or editing a record—created the post.

Warning: If the feed that the process tries to post to isn't available when the process is triggered (for example, because the user is now inactive), the user sees an error and the process fails.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. In the `Post to` field, select User.
3. For `User`, select where you want to find the user.
4. Based on your selection for `User`, search for or browse for the user whose feed you want to post to.

   When you select a user from a record, you must ultimately select a field that contains a user's ID—for example, `Owner ID` or `User ID`.

5. Fill out the message that you want to post. You can insert merge fields, add a topic, and mention users or groups.

   The message can contain up to 10,000 characters.

   You can only reference topics that already exist. If you reference a merge field and that field doesn't have a value, it appears as a blank value.

6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Post to a Chatter Group from a Process

Post to the feed of a specific Chatter group.

After you've created an action and selected "Post to Chatter" for the action type, fill in the relevant fields to add the action to your process. The feed item will appear as if the user who started the process—by creating or editing a record—created the post.

Warning: If the feed that the process tries to post to isn't available when the process is triggered, the user sees an error and the process fails.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. In the `Post to` field, select Chatter Group.
3. For `Group`, search for the Chatter group whose feed you want to post to.
4. Fill out the message that you want to post. You can insert merge fields, add a topic, and mention users or groups.

   The message can contain up to 10,000 characters.

   You can only reference topics that already exist. If you reference a merge field and that field doesn't have a value, it appears as a blank value.

5. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Post to a Record's Chatter Feed from a Process

Post to the feed of the record that started the process.

You can post to the record's Chatter feed only if feed tracking is enabled for the object that the process is associated with. The feed item will appear as if the user who started the process—by creating or editing a record—created the post.

After you've created an action and selected "Post to Chatter" for the action type, fill in the relevant fields to add the action to your process.

Warning: If the feed that the process tries to post to isn't available when the process is triggered (for example, because the user is now inactive), the user sees an error and the process fails.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. In the `Post to` field, select This Record.
3. Fill out the message that you want to post. You can insert merge fields, add a topic, and mention users or groups.

   The message can contain up to 10,000 characters.

   You can only reference topics that already exist. If you reference a merge field and that field doesn't have a value, it appears as a blank value.

4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Use a Quick Action from a Process

Create a record, update a record, or log a call by using a quick action that you or another administrator created for your organization.

Quick actions can be object-specific or global actions. Only "Create," "Update," and "Log a Call" actions are supported. To use a quick action from a process, the action must exist in your organization.

If your organization is using quick actions to help your users more easily create and update records, you can also use those actions in your process. When you use these quick actions in a process, you can only set values for fields that are part of the action's layout. If you don't already have one of these actions created, see Create Global Quick Actions on page 863 or Create Object-Specific Quick Actions on page 866 for details.

After you've created an action and selected "Quick Actions" for the type, fill in the relevant fields to add the action to your process. The new or updated record will appear as if the user who started the process—by creating or editing a record—created or updated it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. Filter to specify the kind of action you want to use.

Filter Search By	Lets You Search Through
Global actions	All global actions in your organization. You then filter even further by selecting the Type of actions that you need to search through.
Object	All object-specific actions in your organization that are associated with a certain `Object Name`. Global actions can't be filtered by object.
Type	All object-specific and global actions in your organization based on the type, such as Create a Record or Log a Call.

   - If you selected Global actions or Type, for `Type` select the specific type of quick action that you want to use.
   - If you selected Object, for `Object` search for and select the object that you want to filter by.

3. For `Action`, search for and select the action that you want to use.
4. Set field values for the action.

   Rows that appear automatically represent the action's required fields. To set values for the action's optional fields, add rows.

`Field`	Select the field whose value you want to set. To filter the drop-down list, type the name of the field. You can set values for fields that are included in the action's layout only.
`Type`	Select the type of value that you want to use. The available types depend on the field that you've selected.
`Value`	Set a value for the field. by using the text entry field to manually enter a value or the field picker to use a field value from a related record. See Setting Values in the Process Builder on page 345 for details.

5. Click **Save**.

Launch a Flow from a Process

Start an autolaunched flow from your process to automate complex business processes—create flows to perform logic and have events trigger the flows via processes—without writing code.

In order to launch a flow from a process, you must create and activate the flow. The flow must be autolaunched. For details, see Create a Flow on page 483.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. For `Flow`, search for and select the flow that you want to launch from this process.

   Only active, autolaunched flows are available.

3. Optionally, click **Add Row** to set values for the flow's variables.

`Flow Variable`	Start typing the name of the flow variable whose value you want to set, or click to select a flow variable from the drop-down list. You can set values for any variables in the flow that have `Input/Output Type` set to Input Only or Input and Output.
`Type`	Select the type of value that you want to set. For example, select **String** to manually enter the values for a Text collection variable, or select **Reference** to use the value of a record for an sObject variable.
`Value`	Set a value for the flow variable. • For collection variables, use the text entry field to specify a value. The value must match the collection variable's data type. • For sObject variables, use the field picker to select an ID field. The ID must correspond to a record whose object type matches the sObject variable's object type. • For sObject collection variables, use the field picker to select a related list. The selected records' object type must match the sObject collection variable's object type. For example, populate an sObject collection variable with all child contact records associated with the account that started the process. Note: If the related list is empty when the flow tries to populate the sObject collection variable with field values from those records, the process fails.

4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Send an Email from a Process

Easily send an email from a process by using an email alert. Email alerts are configured outside of the Process Builder and contain the standard text, list of recipients, and template for an email.

To send an email from a process, you must create the email alert on page 616.

You can use only email alerts that are associated with the same object that the process is associated with. The record that started the process is used as the starting point for any merge fields that are used in the email alert.

After you've created an action and selected "Email Alerts" for the type, fill in the relevant fields to add the action to your process.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. For `Email Alert`, type two or more letters to search for the email alert that you want to use to send an email.
3. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create processes:
- "View All Data"

Submit a Record for Approval from a Process

Submit the record that started the process for approval.

After you've created an action and selected "Submit for Approval" for the type, fill in the relevant fields to add the action to your process.

Only the record that started the process will be submitted. You can't submit any related records for approval.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. For `Approval Process`, indicate whether you need to submit the record through the default approval process or through a specific approval process.

   The process fails if:

   - The record is submitted to the default approval process, and there are no active approval processes for the record's object type.
   - The record is submitted to the default approval process, and it doesn't meet the criteria for any of the approval processes for the record's object type.
   - The record is submitted to a specific approval process, and it doesn't meet the entry criteria.

3. If you need to submit the record to a specific approval process:
   a. Search for and select the approval process.
   b. Indicate whether you need to skip the entry criteria for the approval process.
4. For `Submitter`, identify who should receive notifications about the approval request.

Value	Description
Current User	The user who triggered the process by creating or editing a record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:
- "Manage Force.com Flow"

  AND

  "View All Data"

Value	Description
User Field from a Record	The user ID that's stored in a field value on the record that's being submitted for approval or another record.
Other User	A specific user in your organization.

If the submitter isn't an allowed initial submitter on the approval process that runs, the process fails. Make sure that the initial submitters for the approval processes that are related to this object include all users who could trigger this process. For details about setting the initial submitters for an approval process, see Create an Approval Process with the Standard Wizard on page 411.

5. If necessary, enter submission comments. Don't reference merge fields or formula expressions.

   Submission comments appear in the approval history for the specified record. This text also appears in the initial approval request email if the template uses the `{!ApprovalRequest.Comments}` merge field.

6. Click **Save**.

Update Records from a Process

Update one or more records that are related to the record that started the process by manually entering values or by using the values from related records.

After you've created an action and selected "Update Records" for the action type, fill in the relevant fields to add the action to your process. The records' `Last Modified By` field is set to the user who started the process by creating or editing a record.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. For `Record Type`, select the record or records that you need to update, and then click **Choose**.

   You can update only the record that started the process or records that are related to it. For example, you can reference *`[Case].ContactId`*, but not *`[Case].Contact.AccountId`*.

   - To update the record that started the process, click the appropriate radio button. For example, if your process is based on a case record, click next to **Select the Case record that started your process**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

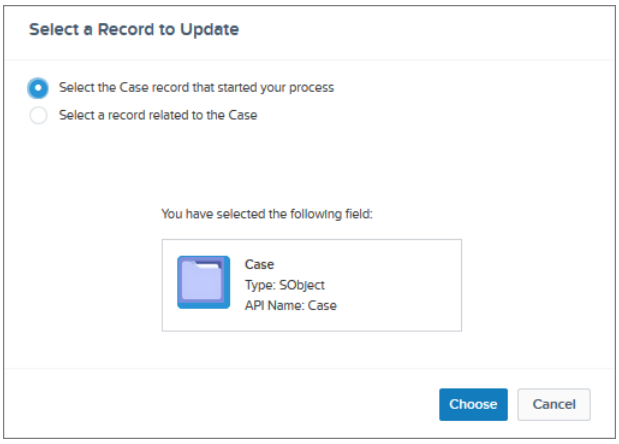
USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

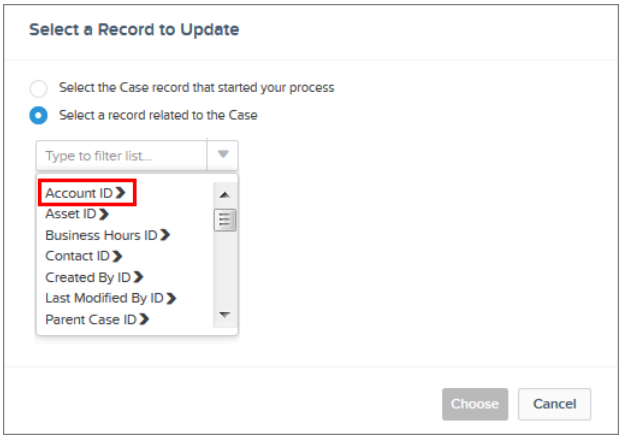
  "View All Data"



- To update a record that's related to the record that started the process, click the appropriate radio button and select one of the field names in the drop-down list.

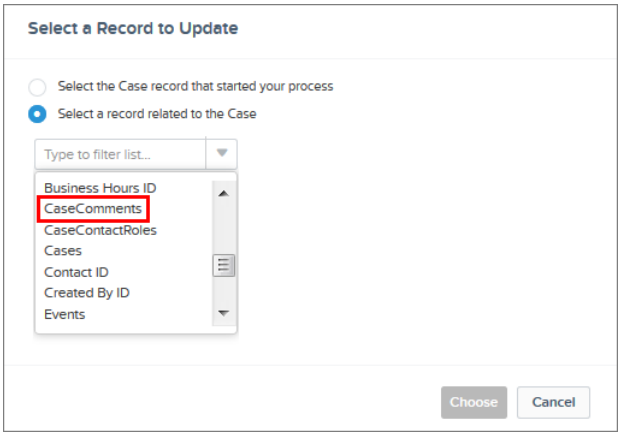
  If you select a field that ends in "ID," you're selecting a single record. This field name corresponds to a lookup field on the original record. For details on lookup fields, see Custom Field Types on page 129.

  For example, if a case record started the process and you select `Account Id`, this action updates the account that's associated with the case.



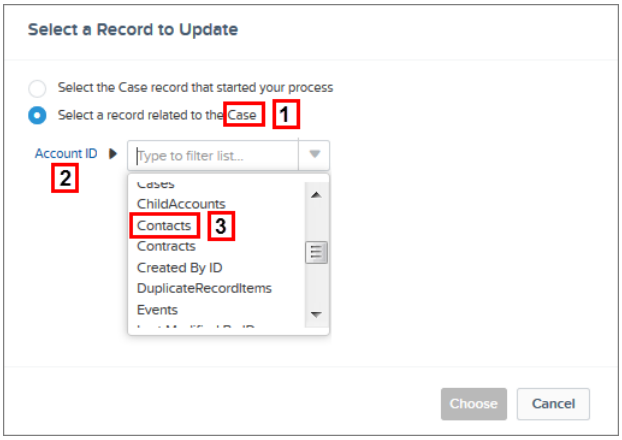
  If you select a plural item that doesn't end in "ID," you're updating all the records of that object type that are related to the record that started the process. This plural item corresponds to child records of the original record, which might appear in a related list on the original record.

  For example, if you select CaseComments, this action updates all the case comments that are related to the case.



- To update fields on a related record, click on a field with ❯ next to it (ending in "ID") to access that record's fields.

  For example, let's say that, for a process that evaluates a case record **(1)**, you want to update all contacts that are related to the case's parent account. Click **Account ID** ❯ **(2)**, then **Contacts (3)**, and then **Choose**.



3. Optionally, specify conditions to filter the records you're updating.

   For example, if your process updates the status of a parent case, specify conditions so that you won't update the parent case if its status is set to On Hold.

   Note: When you define conditions for updating records, you can't:

   - Reference a Long Text Area field
   - Reference a Rich Text field
   - Reference a child record's related fields

     For example, you can reference `[Case].ContactId`, but not `[Case].Contact.AccountId`.

     When you define multiple filters, the filter logic usually defaults to AND. However, if multiple filters have the same field selected and use the equals operator, the filters are combined with OR. For example, your filters check whether a case's Type equals Problem (1), Type equals Feature Request (2), and Escalated equals true (3). At runtime, those filters are combined to `(1 OR 2) AND 3`.

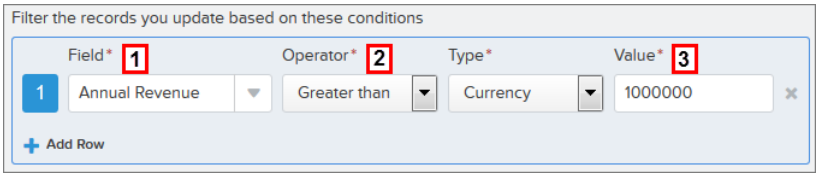
If you're updating the record that started the process, Process Builder adds an implicit filter for you in the background: `[Object].Id equals myCurrentVariable.Id`. If you add filter criteria that set the record's ID to a value using the equals operator, at runtime the `[Object].Id equals` filters are combined using OR filter logic. For example, you update the case that started the process and add this filter: `[Case].Id equals 500D00000044XgV`. At runtime, your filter is combined with the implicit filter (`[Case].Id equals myCurrentVariable.Id`) with OR.

a. Select **Updated records meet all conditions**.

b. Set the conditions that you want to use to filter the updated records.

`Field`	Select the field whose value you want to evaluate.
`Operator`	The available operators depend on the field's data type.
`Type`	The available value types depend on the field' data type. See Process Builder Value Types on page 347 for details.
`Value`	Identify the value that you want to evaluate the field for.

For example, if your process updates account records, you can choose to update only accounts with an annual revenue (**1**) greater than (**2**) $1,000,000 (**3**).



4. Specify the new field values.

`Field`	Select the field whose value you want to set. To filter the drop-down list, type the name of the field. You can assign values to fields only on the record or records that you identified in the `Object` field. Use a separate Update Records action to update fields on related records.
`Type`	Select the type of value that you want to use. The available types depend on the field that you've selected.
`Value`	Set a value for the field. For example, if you select a Formula value type, click **Build a formula...** to create a formula value for the field.

5. Click **Save**.

Call Apex Code from a Process

Add customized functionality to your process by calling Apex from the process.

After you've created an action and selected "Apex" for the type, fill in the relevant fields to add the action to your process.

Important: To use this action in a process, your developer needs to annotate one of the class's methods with `@InvocableMethod`. For details, see "`InvocableMethod` Annotation" in the *Force.com Apex Code Developer's Guide*.

The Apex class and the process are executed by the user whose action triggered the process.

1. Enter a name for this action.

   This text appears on the canvas and helps you differentiate this action from others in your process. The name is truncated to fit on the canvas.

2. Choose an Apex class by entering the name of the Apex class to filter results or select an Apex class from the drop-down list.

3. If the class includes an invocable variable, you can manually enter values or reference field values from a related record.

   The value must match the variable's data type. You can set values for sOjbect and primitive type list variables only.

   - To set values for sObject variables and sObject list values, reference an object's related records, for example, all child contact records associated with the Account object that started the process.
   - To set a value for a primitive list variable (String, Integer, Time, and so on), select the String value type and enter a value in the text input field. You can't pass multiple values to lists.

4. Click **Save**.

Note: If you define an Apex action in your process and then modify the Apex class by adding a standard field reference (for example, `User.Phone`), the Apex action is no longer visible in the process and must be re-added.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

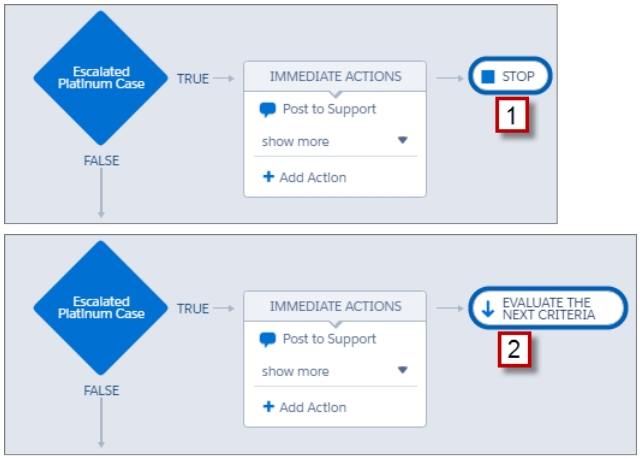
  "View All Data"

Execute Actions on More Than One Criteria

Choose whether to stop or continue your process after specific criteria are met and associated actions execute.

For each criteria and associated action group, you can choose whether to stop the process after actions execute or continue evaluating the next defined criteria.

1. Make sure you've defined the next criteria and that your action group includes only immediate actions. You can't evaluate the next criteria when an action group contains scheduled actions.
2. To change what happens after actions execute, click **STOP (1)** or **EVALUATE THE NEXT CRITERIA (2)**. Initially, each action group is set to stop after executing actions.



3. Save your changes, and your choice appears on the canvas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, edit, or view processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Process Management

The Process Builder allows you to see and manage all your processes in one place.

To manage a process, from Setup, enter *Process Builder* in the Quick Find box, then select **Process Builder**.

From the process management page, you can:

- Create a process
- Edit a process
- Delete an inactive process
- See the status of your existing processes
- Sort your processes by name, description, object, last modified date, or status

Note: If you create a Salesforce Knowledge Article Type object and use that object to trigger a process, the object name doesn't display on the process management page.

When you open a process, you can:

- Clone the process
- Activate or deactivate the process
- Edit the process properties

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To manage processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

From the Paused and Waiting Interviews list on the flow management page, you can:

- Monitor scheduled actions that haven't yet been executed
- Delete groups of scheduled actions that you no longer need to wait for

Process Status

Each process has a status that determines whether the process can be edited, activated, or deleted.

Status	Description	Editable?
Active	The process has been activated. You can't make changes to an active process. However, you can clone it. Make any necessary changes to the cloned process and then activate it. Don't forget to deactivate the original process if appropriate.	No
Inactive	The process is inactive and can be activated.	Yes

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Clone a Process

If you want to make changes to an existing process, save a clone of that process. You can save the clone as either a new inactive process with its own version history, or as a new inactive version of the existing process.

To make changes to an active process, you have a few options.

- Deactivate it, make changes, and then reactivate it.
- Clone it as an inactive version, make changes, and then activate the new version. The original version is automatically deactivated.
- Clone it as a new inactive process, make changes and then activate it. The original process is not automatically deactivated, so consider whether it's appropriate for both processes to be active.

You can create up to 50 versions of a process, but only one version of a given process can be active.

1. From Setup, enter *Process Builder* in the Quick Find box, then select **Process Builder**.
2. Open the process or process version that you want to activate.
3. Click **Clone**.
4. You can create a version of the current process or a completely new process with its own version history.
5. Enter a name, API name, and description.
6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To activate or deactivate processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

Activate a Process

Salesforce doesn't start using a new or revised process to evaluate records until you activate it.

After you activate a process, you can no longer edit it. However, you can click **Clone** to save the process as a new inactive process.

You can't activate a process unless it has:

- At least one defined criteria node
- At least one defined immediate or scheduled action

1. From Setup, enter *Process Builder* in the Quick Find box, then select **Process Builder**.
2. Open the process or process version that you want to activate.
3. Click **Activate**.

   If you activate a version of a process that already has an active version, the previously active version is automatically deactivated. To see that version later, refer to the process's version history.

After you've activated your process, consider creating or editing test records that will start the process to make sure it's working correctly. If you do, remember to delete those test records or return them to their previous values once you've confirmed that your process works as designed.

If you later want Salesforce to stop using a process to evaluate records as they're created or edited, open the active process and click **Deactivate**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To activate or deactivate processes:

- "Manage Force.com Flow"

  AND

  "View All Data"

  AND

  "Customize Application"

Delete a Process Version

If you no longer need to use a process version that you've defined, delete it.

To delete an active process, you must first deactivate it. You can't delete process versions with an Active status. If another process references your invocable process, you can't delete the invocable process. If a process has any scheduled actions, it can't be deleted until those pending actions have been executed or deleted.

1. From Setup, enter *Process Builder* in the Quick Find box, then select **Process Builder**.
2. Click the name of the process whose version you want to delete.
3. For the version that you want to delete, click **Delete**.

   If your process has only one version and you delete that version, the whole process is deleted.
4. Click **OK**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To delete processes:

- "Manage Force.com Flow"

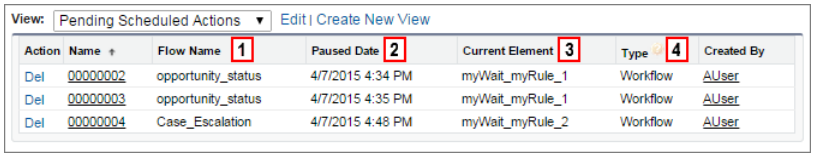
  AND

  "View All Data"

Monitor Your Processes' Pending Scheduled Actions

To see which of your processes are still waiting to execute scheduled actions, see the Paused and Waiting Interviews list on the flow management page.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.
2. Scroll down to the Paused and Waiting Interviews list.
3. Create a view to see scheduled actions in the Paused and Waiting Interviews list. We recommend displaying the following fields.



- Flow Name **(1)** corresponds to the process's Process Name.
- Paused Date **(2)** is the time at which the schedule started for the action group.
- Current Element **(3)** identifies the group of scheduled actions that Salesforce is waiting to execute.
- Type **(4)** indicates the type of a paused or waiting interview. Processes that are waiting to execute scheduled actions are of type Workflow.

Current Element displays myWait_myRule_*N*, where *N* is the number of the associated criteria and action group. For example, if Current Element displays myWait_myRule_2, Salesforce is waiting to execute the group of scheduled actions that are associated with the second criteria in the process.

SEE ALSO:

Delete Unexecuted Scheduled Actions

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To see unexecuted scheduled actions:

- "View Setup and Configuration"

Delete Unexecuted Scheduled Actions

If you no longer need to wait for a process's scheduled actions to be executed—for example, because you need to delete the associated process—you can delete the unexecuted group of scheduled actions.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.

   The Waiting Interviews related list displays any unexecuted scheduled actions in your organization, as well as any waiting flow interviews.

2. For each unexecuted group of scheduled actions that you want to delete, click **Del**.

   You can identify the correct row by the Flow Name field, which corresponds to the process's Process Name.

SEE ALSO:

Monitor Your Processes' Pending Scheduled Actions

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To delete unexecuted scheduled actions:

- "Manage Force.com Flow"

Troubleshoot Processes

Use the error messages that appear in the Process Builder and the emails you receive when a process fails to help solve problems that arise when you're working with processes. When all else fails, look at the Apex debug logs for your processes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Identifying Errors in the Process Builder

The API names for criteria nodes and actions are created in the background. When you create or update processes, you might see error messages that reference those names to help you identify specifically where the problem occurred.

Errors Received after a Process Starts Evaluating a Record

When a user performs an action that triggers a process (such as creating a record) and that process fails, the user sees a page with this error: "Workflow Action Failed to Trigger Flow." In addition, the administrator who activated the process receives an email with more details.

Using Debug Logs to Troubleshoot Processes

Use debug logs to find detailed information about your running processes after they finish running. For example, if a process doesn't seem to trigger when a record meets the process's criteria, or if you want to understand the sequence of processes being executed.

SEE ALSO:

Process Builder Error Messages

Identifying Errors in the Process Builder

The API names for criteria nodes and actions are created in the background. When you create or update processes, you might see error messages that reference those names to help you identify specifically where the problem occurred.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

API Name	Description
`myVariable_current.`*`field`*	*`field`* is the name of the field that's referenced. `myVariable_current` refers to the field values that the record had when it started the process. For example, `myVariable_current.Id` corresponds to the record's field value for `Id` when the record started the process.
`myVariable_old.`*`field`*	*`field`* is the name of the field that's referenced. `myVariable_old` refers to the most recent previous values of the record that started the process. For example, `myVariable_old.Id` corresponds to the record's field value for `Id` immediately before the record started the process.

Example:

```
The element has an invalid reference to "myVariable_current.AnnualRevenue".
```

`myVariable_current.AnnualRevenue` refers to the value for the field `AnnualRevenue` when the record started the process.

Note: Error or warning messages might refer to a "flow" instead of a "process." Those messages still apply to your process.

SEE ALSO:

Process Builder Error Messages

Errors Received after a Process Starts Evaluating a Record

When a user performs an action that triggers a process (such as creating a record) and that process fails, the user sees a page with this error: "Workflow Action Failed to Trigger Flow." In addition, the administrator who activated the process receives an email with more details.

The email includes the element that failed, the error message from that failure, and a list of the criteria and actions that were executed in the process. The subject line is `Unhandled process fault from` *`Company Name`* `:` *`Process_Name`* `---` *`Error`*.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example:

```
Unhandled process fault from Acme Wireless : LeadConvert :
cannot reference converted...
// The error occurred when user 00DD00000007dHy triggered the
 LeadConvertEmail process
Encountered unhandled fault when running process
LeadConvertEmail/301D00000004HxZ
exception by user/organization: 00DD00000007dHy/{4}

Error
// The error occurred at the first action (A1) that's
associated with the
// first criteria node (myRule_1).
An error occurred at element myRule_1_A1.
cannot reference converted lead .

This report lists the elements that the flow interview
executed.

Flow Details
Flow Name: LeadConvertEmail
Type: Workflow
Version: 1
Status: ACTIVE

Flow Details
Start time(GMT): Thu Apr 09 21:52:14 GMT 2015
End time(GMT): Not Yet Finished
Duration: 0 seconds
```

```
// The process evaluated the first rule and tried to execute its action. The element
// that the process failed to executed, myRule_1_A1, is an Email Alert action.
Decision: myDecision
Email Alerts: myRule_1_A1
```

SEE ALSO:

Process Builder Error Messages

Using Debug Logs to Troubleshoot Processes

Use debug logs to find detailed information about your running processes after they finish running. For example, if a process doesn't seem to trigger when a record meets the process's criteria, or if you want to understand the sequence of processes being executed.

Tip: Make sure that your filters include FINER events in the WORKFLOW category. For details, see Debug Log Levels.

When using debug logs to troubleshoot a process, consider the following.

- Processes created in the Process Builder appear as flows and workflow rules in debug logs. The generated names have some resemblance to the process names, but they don't map one-to-one.
- Processes appear as flows of type Workflow.
- Immediate actions that may appear to be executed in a block are strung together in the flow. If one action fails in the middle, then the following actions might not run.
- Scheduled actions are executed after a Wait element. The actions are executed similarly to immediate actions after the process resumes.
- `WF_CRITERIA_BEGIN` and `WF_CRITERIA_END` refer to the workflow rule criteria that is always set to true and not the criteria defined in your process.
- Here's how Process Builder elements correspond to flow debug events.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Process Builder Element	Flow Debug Event
Criteria	`FLOW_RULE_...`
Create a Record	`FLOW_ELEMENT_...`
Update Records	`FLOW_ELEMENT_...`
Post to Chatter	`FLOW_ACTIONCALL_...`
Submit for Approval	`FLOW_ACTIONCALL_...`
Email Alerts	`FLOW_ACTIONCALL_...`
Flows	`FLOW_ACTIONCALL_...`
Apex	`FLOW_ACTIONCALL_...`
Schedule	`FLOW_WAIT_...`

Example: **Debugging a Process with an Immediate "Apex" Action**

This example covers a process with an immediate "Apex" action.

Here's what you can tell from this first snippet.

- The process is triggered by a lead with the name "L104".
- The name of the process is "Apex Say Hello", which corresponds with the name used in the log "Apex_Say_Hello301xx000000001Y". The number appended to the end of the name is the process version's ID.
- The process is set to trigger when a record is created or edited (ON_ALL_CHANGES).

```
17:07:29.086 (86806000)|EXECUTION_STARTED
17:07:29.086 (86928000)|CODE_UNIT_STARTED|[EXTERNAL]|Workflow:Lead
17:07:29.113 (113974000)|WF_RULE_EVAL_BEGIN|Assignment
17:07:29.114 (114043000)|WF_SPOOL_ACTION_BEGIN|Assignment
17:07:29.114 (114060000)|WF_ACTION|.
17:07:29.114 (114071000)|WF_RULE_EVAL_END
17:07:29.114 (114093000)|WF_RULE_EVAL_BEGIN|Workflow
17:07:29.114 (114178000)|WF_CRITERIA_BEGIN|
   [Lead: L104 00Qxx000002S8vh]|Apex_Say_Hello301xx000000001Y|
   01Qxx0000002nea|ON_ALL_CHANGES|0
```

In this section, the process compares the record's current values to the values it had before it was changed. "myVariable_current" contains all of the record's current field values. "myVariable_old" contains all of the field values of the record immediately before it was changed. "myVariable_old" has no values (null), so the process is evaluating a newly created lead.

```
17:07:29.114 (114553000)|WF_FORMULA|
   Formula:ENCODED:[treatNullAsNull]true|Values:
17:07:29.114 (114571000)|WF_CRITERIA_END|
   true
17:07:29.132 (132792000)|WF_SPOOL_ACTION_BEGIN|
   Workflow
17:07:29.132 (132881000)|WF_ACTION|
   Flow Trigger: 1;
17:07:29.132 (132897000)|WF_RULE_EVAL_END
17:07:29.137 (137127000)|WF_FLOW_ACTION_BEGIN|
   09Lxx000000000B
17:07:29.137 (137249000)|WF_FLOW_ACTION_DETAIL|
   09Lxx000000000B|[Lead: L104 00Qxx000002S8vh]|Id=09Lxx000000000B|
   CurrentRule:Apex_Say_Hello301xx000000001Y (Id=01Qxx0000002nea)
17:07:29.139 (139429000)|FLOW_CREATE_INTERVIEW_BEGIN|
   00Dxx0000001gH6|300xx00000002cI|301xx000000001Y
17:07:29.140 (140996000)|FLOW_CREATE_INTERVIEW_END|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|Apex Say Hello 2
17:07:29.142 (142898000)|WF_FLOW_ACTION_DETAIL|Param Name: myVariable_current,
   Param Value: ENCODED:{![treatNullAsNull]{!ID:this}},
   Evaluated Param Value:{
      LastModifiedByID=005xx000001SwUpAAK, LastTransferDate=2014-12-12 01:07:29,
     ConnectionReceivedDate=null, LastModifiedDate=2014-12-12 01:07:29, LeadSource=null,

      LastName=L104, NumberofLocations__c=null, NumberOfEmployees=null,
DandbCompanyId=null,
      DoNotCall=false, Primary__c=null, CreatedByID=005xx000001SwUpAAK,
ConnectionSentDate=null,
      MobilePhone=null, Jigsaw=null, Salutation=null, HasOptedOutOfEmail=false,
```

```
      Country=null, LastViewedDate=null, ConvertedOpportunityId=null,
EmailBouncedReason=null,
      Email=null, Rating=null, SICCode__c=null, CleanStatus=Pending,
CreatedById=005xx000001SwUpAAK,
     IsDeleted=false, Longitude=null, Id=00Qxx000002S8vhEAC, Industry=null, Street=null,

      Phone=null, IsConverted=false, EmailBouncedDate=null, HasOptedOutOfFax=false,
      OwnerId=005xx000001SwUpAAK, AnnualRevenue=null, CompanyDunsNumber=null,
IsUnreadByOwner=true,
     LastActivityDate=null, SystemModstamp=2014-12-12 01:07:29, ProductInterest__c=null,

      Latitude=null, FirstName=null, PhotoUrl=null, Company=LC,
CampaignMemberStatus=null,
      Description=null, LastReferencedDate=null, LastModifiedById=005xx000001SwUpAAK,

      Website=null, Fax=null, InternalSource=0, City=null, Title=null,
MasterRecordId=null,
      CampaignId=null, Status=Open - Not Contacted, JigsawContactId=null,
ConvertedDate=null,
      PostalCode=null, State=null, CreatedDate=2014-12-12 01:07:29,
ConvertedContactId=null,
      ConvertedAccountId=null, CurrentGenerators__c=null
   }|
   Param Name: myVariable_old,
   Param Value: {!old},
   Evaluated Param Value: null
```

This section shows the start of the process—indicated by "FLOW_START_INTERVIEWS_BEGIN"—and the criteria defined in the process. "FLOW_RULE_" always corresponds to criteria in the process, and we see that "myRule_1" is evaluated as "true". The number in "myRule_1" indicates that the process is executing the group of scheduled actions that are associated with the first criteria in the process.

```
17:07:29.143 (143184000)|FLOW_START_INTERVIEWS_BEGIN|1
17:07:29.147 (147213000)|FLOW_START_INTERVIEW_BEGIN|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|Apex Say Hello 2
17:07:29.147 (147389000)|FLOW_VALUE_ASSIGNMENT|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myVariable_old|
17:07:29.148 (148138000)|FLOW_VALUE_ASSIGNMENT|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myVariable_current|{
      LastModifiedByID=005xx000001SwUpAAK, LastTransferDate=2014-12-12 01:07:29,
     ConnectionReceivedDate=null, LastModifiedDate=2014-12-12 01:07:29, LeadSource=null,

      LastName=L104, NumberofLocations__c=null, NumberOfEmployees=null,
DandbCompanyId=null,
      DoNotCall=false, Primary__c=null, CreatedByID=005xx000001SwUpAAK,
ConnectionSentDate=null,
      MobilePhone=null, Jigsaw=null, Salutation=null, HasOptedOutOfEmail=false,
Country=null,
      LastViewedDate=null, ConvertedOpportunityId=null, EmailBouncedReason=null,
Email=null,
     Rating=null, SICCode__c=null, CleanStatus=Pending, CreatedById=005xx000001SwUpAAK,

     IsDeleted=false, Longitude=null, Id=00Qxx000002S8vhEAC, Industry=null, Street=null,
```

```
      Phone=null, IsConverted=false, EmailBouncedDate=null, HasOptedOutOfFax=false,
      OwnerId=005xx000001SwUpAAK, AnnualRevenue=null, CompanyDunsNumber=null,
IsUnreadByOwner=true,
     LastActivityDate=null, SystemModstamp=2014-12-12 01:07:29, ProductInterest__c=null,

      Latitude=null, FirstName=null, PhotoUrl=null, Company=LC,
CampaignMemberStatus=null,
      Description=null, LastReferencedDate=null, LastModifiedById=005xx000001SwUpAAK,
 Website=null,
      Fax=null, InternalSource=0, City=null, Title=null, MasterRecordId=null,
CampaignId=null,
      Status=Open - Not Contacted, JigsawContactId=null, ConvertedDate=null,
PostalCode=null,
      State=null, CreatedDate=2014-12-12 01:07:29, ConvertedContactId=null,
ConvertedAccountId=null,
      CurrentGenerators__c=null
   }

17:07:29.148 (148427000)|FLOW_ELEMENT_BEGIN|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|FlowDecision|myDecision
17:07:29.149 (149938000)|FLOW_RULE_DETAIL|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myRule_1|true
17:07:29.149 (149976000)|FLOW_VALUE_ASSIGNMENT|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myRule_1|true
17:07:29.150 (150050000)|FLOW_ELEMENT_END|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|FlowDecision|myDecision
```

In this section, the immediate actions for the first criteria are executed. In the name "myRule_1_A1", "A1" indicates that this corresponds to the first action in the action group. We also see the Apex class "SayHello.sayHello" and the details of the Apex class execution. Then the process finishes with no errors.

```
17:07:29.150 (150117000)|FLOW_ELEMENT_BEGIN|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|FlowActionCall|myRule_1_A1
17:07:29.150 (150146000)|FLOW_ELEMENT_DEFERRED|
   FlowActionCall|myRule_1_A1
17:07:29.150 (150194000)|FLOW_ELEMENT_END|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|FlowActionCall|myRule_1_A1
17:07:29.150 (150233000)|FLOW_START_INTERVIEW_END|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|Apex Say Hello 2

17:07:29.150 (150316000)|FLOW_BULK_ELEMENT_BEGIN|
   FlowActionCall|myRule_1_A1
17:07:29.151 (151772000)|CODE_UNIT_STARTED|
   [EXTERNAL]|01pxx00000000DN|SayHello.sayHello
17:07:29.153 (153773000)|METHOD_ENTRY|
   [1]|01pxx00000000DN|SayHello.SayHello()
17:07:29.153 (153821000)|METHOD_EXIT|
   [1]|SayHello
17:07:29.154 (154292000)|SYSTEM_CONSTRUCTOR_ENTRY|
   [4]|<init>()
17:07:29.154 (154392000)|SYSTEM_CONSTRUCTOR_EXIT|
   [4]|<init>()
17:07:29.154 (154564000)|SYSTEM_METHOD_ENTRY|
   [5]|List<String>.add(Object)
```

```
17:07:29.154 (154748000)|SYSTEM_METHOD_EXIT|
   [5]|List<String>.add(Object)
17:07:29.084 (154888000)|CUMULATIVE_LIMIT_USAGE
17:07:29.084|LIMIT_USAGE_FOR_NS|(default)|
  Number of SOQL queries: 0 out of 100
  Number of query rows: 0 out of 50000
  Number of SOSL queries: 0 out of 20
  Number of DML statements: 0 out of 150
  Number of DML rows: 0 out of 10000
  Maximum CPU time: 0 out of 10000
  Maximum heap size: 0 out of 6000000
  Number of callouts: 0 out of 100
  Number of Email Invocations: 0 out of 10
  Number of future calls: 0 out of 50
  Number of queueable jobs added to the queue: 0 out of 50
  Number of Mobile Apex push calls: 0 out of 10

17:07:29.084|CUMULATIVE_LIMIT_USAGE_END

17:07:29.154 (154949000)|CODE_UNIT_FINISHED|
   SayHello.sayHello
17:07:29.155 (155090000)|FLOW_ACTIONCALL_DETAIL|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myRule_1_A1|Call Apex|SayHello|true|
17:07:29.155 (155133000)|FLOW_VALUE_ASSIGNMENT|
   108a970cb7aeabf8804183a600514a3bb92516-7f92|myRule_1_A1|true
17:07:29.155 (155268000)|FLOW_BULK_ELEMENT_END|
   FlowActionCall|myRule_1_A1|0|4

17:07:29.155 (155371000)|FLOW_START_INTERVIEWS_END|
   1
17:07:29.159 (159563000)|WF_FLOW_ACTION_END|
   09Lxx000000000B
17:07:29.159 (159787000)|WF_ACTIONS_END|
   Flow Trigger: 1;
17:07:29.159 (159815000)|CODE_UNIT_FINISHED|
   Workflow:Lead
17:07:29.161 (161221000)|EXECUTION_FINISHED
```

Example: **Debugging a Process with a Scheduled "Post to Chatter" Action**

This example highlights parts of a debug log for a process with a scheduled "Post to Chatter" action.

A schedule is always indicated by "myWait_". This section shows the process's running schedule "myWait_myRule_2". Any events that start with "FLOW_WAIT_" also correspond to a schedule in a process.

```
20:06:28.928 (928266000)|FLOW_ELEMENT_BEGIN|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|WaitInfo|myWait_myRule_2
20:06:28.928 (928302000)|FLOW_ELEMENT_DEFERRED|
   WaitInfo|myWait_myRule_2
20:06:28.928 (928361000)|FLOW_ELEMENT_END|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|WaitInfo|myWait_myRule_2
20:06:28.929 (929671000)|FLOW_BULK_ELEMENT_BEGIN|
   WaitInfo|myWait_myRule_2
20:06:29.005 (1005229000)|FLOW_WAIT_EVENT_WAITING_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myWait_myRule_2|
```

```
   myWaitEvent_myRule_2|alarmDateRef|true
20:06:29.005 (1005317000)|FLOW_VALUE_ASSIGNMENT|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myWaitEvent_myRule_2|true
20:06:30.225 (2225117000)|FLOW_WAIT_WAITING_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myWait_myRule_2|1|0Foxx0000000001
20:06:30.225 (2225362000)|FLOW_BULK_ELEMENT_END|
   WaitInfo|myWait_myRule_2|0|1296
```

...

```
20:06:30.234 (2234916000)|WF_TIME_TRIGGERS_BEGIN
20:06:30.248 (2248007000)|WF_ACTIONS_END|
   Flow Trigger: 1;
20:06:30.248 (2248072000)|CODE_UNIT_FINISHED|
   Workflow:Lead
20:06:30.250 (2250129000)|EXECUTION_FINISHED
```

For scheduled actions, the debug log shows an additional log when the scheduled actions are performed. This is a separate entry in the debug log, as shown in the following example for this scheduled "Post to Chatter" action. After the scheduled time occurs, an automated process executes the scheduled actions. The log uses coordinated universal time (UTC) instead of the user's time zone. However, the actions are still executed as the user who originally caused the process to run.

"FLOW_WAIT_EVENT_RESUMING_DETAIL" indicates that the interview is being resumed so that the process can execute its scheduled actions. "alarmDateRef" indicates that this schedule is based on a field value. We then see that the "Post to Chatter" action is executed.

```
00:14:10.466 (1418357225466561000)|FLOW_BULK_ELEMENT_BEGIN|
   WaitInfo|myWait_myRule_2
00:14:10.469 (1418357225469327000)|FLOW_WAIT_EVENT_RESUMING_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myWait_myRule_2|
   myWaitEvent_myRule_2|alarmDateRef
00:14:10.469 (1418357225469412000)|FLOW_WAIT_RESUMING_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myWait_myRule_2|
   0Foxx000000000100:14:10.469 (1418357225469733000)|
   FLOW_BULK_ELEMENT_END|WaitInfo|myWait_myRule_2|0|3
00:14:10.469 (1418357225469826000)|FLOW_ELEMENT_BEGIN|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|
   FlowDecision|myPostWaitDecision_myRule_2
00:14:10.471 (1418357225471235000)|FLOW_RULE_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myPostWaitRule_myRule_2|true
00:14:10.471 (1418357225471275000)|FLOW_VALUE_ASSIGNMENT|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myPostWaitRule_myRule_2|true
00:14:10.471 (1418357225471348000)|FLOW_ELEMENT_END|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|
   FlowDecision|myPostWaitDecision_myRule_2
00:14:10.471 (1418357225471441000)|FLOW_ELEMENT_BEGIN|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|
   FlowActionCall|myRule_2_SA1
00:14:10.471 (1418357225471486000)|FLOW_ELEMENT_DEFERRED|
   FlowActionCall|myRule_2_SA1
00:14:10.471 (1418357225471551000)|FLOW_ELEMENT_END|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|
   FlowActionCall|myRule_2_SA1
00:14:10.471 (1418357225471624000)|FLOW_BULK_ELEMENT_BEGIN|
```

```
   FlowActionCall|myRule_2_SA1
00:14:10.841 (1418357225841584000)|FLOW_ACTIONCALL_DETAIL|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|
   myRule_2_SA1|Post to Chatter|chatterPost|true|
   00:14:10.841 (1418357225841661000)|FLOW_VALUE_ASSIGNMENT|
   345ad8ae889702ab70b5df5712b14a3c990e9d-7ea5|myRule_2_SA1|true
00:14:10.841 (1418357225841773000)|FLOW_BULK_ELEMENT_END|
   FlowActionCall|myRule_2_SA1|0|370
```

SEE ALSO:

Troubleshoot Processes

Workflow

Workflow lets you automate standard internal procedures and processes to save time across your org. A *workflow rule* is the main container for a set of workflow instructions. These instructions can always be summed up in an if/then statement.

For example: **If it's raining, then bring an umbrella.**

Tip: Whenever possible, automate your if/then statements with Process Builder instead of workflow rules.

Workflow rules can be broken into two main components.

1. Criteria: the "if" part of the "if/then" statement. In other words, what must be true of the record for the workflow rule to execute the associated actions.
2. Actions: the "then" part of the "if/then" statement. In other words, what to do when the record meets the criteria.

In the raining example, the criteria is "it's raining" and the action is "bring an umbrella". If the criteria isn't met (it isn't raining), then the action isn't executed (you don't bring an umbrella).

When a record meets all the criteria for a workflow rule, that rule's actions are executed. Familiarize yourself with the automated actions that are available for workflow.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Create a Workflow Rule

Automate your organization's standard process by creating a workflow rule.

Workflow Considerations

Learn the intricacies of workflow rules and workflow actions before you begin working with them.

Workflow Rule Examples

Looking for ideas on how workflow rules can help streamline your business? Check out these examples.

Monitor Pending Workflow Actions

When a workflow rule that has time-dependent actions is triggered, use the workflow queue to view pending actions and cancel them if necessary.

Workflow Terminology

The following terms are used when describing workflow features and functionality.

SEE ALSO:

Which Automation Tool Do I Use?

Create a Workflow Rule

Automate your organization's standard process by creating a workflow rule.

Tip: Whenever possible, automate your if/then statements with Process Builder instead of workflow rules.

Watch a Demo: Creating a Workflow Rule

IN THIS SECTION:

1. Set the Criteria for Your Workflow Rule
   Get started with creating a new workflow rule by selecting the object the rule relates to and configuring its criteria.
2. Add Automated Actions to Your Workflow Rule
   Once you've set the criteria for your workflow rule, identify what to do when that criteria are met.
3. Identify Your Salesforce Org's Default Workflow User
   Select a `Default Workflow User` that you want Salesforce to display with a workflow rule when the user that triggered the rule is not active.
4. Activate Your Workflow Rule
   Salesforce doesn't trigger a workflow rule until you activate it.

SEE ALSO:

Workflow Considerations

Workflow Rule Examples

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:

- "Customize Application"

Set the Criteria for Your Workflow Rule

Get started with creating a new workflow rule by selecting the object the rule relates to and configuring its criteria.

1. From Setup, enter *Workflow Rules* in the Quick Find box, then select **Workflow Rules**.
2. Click **New Rule**.
3. Choose the object to which you want this workflow rule to apply.
4. Click **Next**.
5. Give the rule a name and description.
6. Set the evaluation criteria.

Evaluate the rule when a record is:	Description
created	Evaluate the rule criteria each time a record is created. If the rule criteria is met, run the rule. Ignore all updates to existing records. With this option, the rule never runs more than once per record.
created, and every time it's edited	Evaluate the rule criteria each time a record is created or updated. If the rule criteria is met, run the rule. With this option, the rule repeatedly runs every time a record is edited as long as the record meets the rule criteria. Note: If you select this option, you can't add time-dependent actions to the rule.
created, and any time it's edited to subsequently meet criteria	(Default) Evaluate the rule criteria each time a record is created or updated. • For a new record, run the rule if the rule criteria is met. • For an updated record, run the rule only if the record is changed from not meeting the rule criteria to meeting the rule criteria. With this option, the rule can run multiple times per record, but it won't run when the record edits are unrelated to the rule criteria. For example, suppose that for an opportunity record to meet the rule criteria, the opportunity probability must be greater than 50%. If you create an opportunity with a probability of 75%, the workflow rule runs. If you edit that opportunity by changing the probability to 25%, the edit doesn't cause the rule to run. If you then edit that opportunity by changing the probability from 25% to 75%, the edit causes the rule to run. With this last edit, the rule runs, because the record is changed from not meeting the rule criteria to meeting the rule criteria.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:
- "Customize Application"

7. Enter your rule criteria.
   - Choose `criteria are met` and select the filter criteria that a record must meet to trigger the rule. For example, set the filter to "Opportunity: Amount greater than 5000" if you want opportunity records with an amount greater than $5,000 to trigger the rule. If your organization uses multiple languages, enter filter values in your individual language. You can add up to 25 filter criteria, of up to 255 characters each.
   - Choose `formula evaluates to true` and enter a formula that returns a value of "True" or "False." Salesforce triggers the rule if the formula returns "True."

     Examples of useful workflow formulas include:

     - If the number of filled positions equals the number of total positions on a job, update the `Job Status` field to "Filled."
     - If mileage expenses associated with visiting a customer site are 35 cents per mile and exceed a $1,000 limit, automatically update the `Approval Required` field to "Required."
     - If a monthly subscription-based opportunity amount is greater than $10,000, create a task for an opportunity owner to follow up 60 days after the opportunity is closed.

     The `$Label` variable isn't supported in workflow rule formulas. Also, some functions aren't available in workflow rule formulas.

     Tip: You can use merge fields for directly related objects in workflow rule formulas.

8. Click **Save & Next**.

SEE ALSO:

Workflow Considerations

Add Automated Actions to Your Workflow Rule

Once you've set the criteria for your workflow rule, identify what to do when that criteria are met.

IN THIS SECTION:

Add an Immediate Action to Your Workflow Rule

Immediate actions, like their name suggests, are executed as soon as the workflow rule finishes evaluating the record.

Add a Time-Dependent Action to Your Workflow Rule

Time-dependent actions are executed at a specific time, such as 10 days before a record's close date. When that specific time passes, the workflow rule re-evaluates the record to make sure that it still meets the rule criteria. If the record does, the workflow rule executes those actions.

SEE ALSO:

Identify Your Salesforce Org's Default Workflow User

Set the Criteria for Your Workflow Rule

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Email alert actions and outbound message actions available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:

- "Customize Application"

Add an Immediate Action to Your Workflow Rule

Immediate actions, like their name suggests, are executed as soon as the workflow rule finishes evaluating the record.

1. Open a workflow rule.
2. In the Immediate Workflow Actions section, click **Add Workflow Action**.
3. Select one of the options to create an action or select an existing one.

For details on each action type, see Automated Actions.

SEE ALSO:

Add Automated Actions to Your Workflow Rule

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:

- "Customize Application"

Add a Time-Dependent Action to Your Workflow Rule

Time-dependent actions are executed at a specific time, such as 10 days before a record's close date. When that specific time passes, the workflow rule re-evaluates the record to make sure that it still meets the rule criteria. If the record does, the workflow rule executes those actions.

Time-dependent actions and time triggers are complex features. As you work with time-dependent actions and time triggers, keep in mind their considerations.

If you plan on configuring workflow rules that have time-dependent actions, specify a default workflow user. Salesforce associates the default workflow user with a workflow rule if the user who initiated the rule is no longer active.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:
- "Customize Application"

1. Open a workflow rule.
2. In the Time-Dependent Workflow Actions section, click **Add Time Trigger**.

   Note: You can't add a time trigger if:
   - The evaluation criteria is set to Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria.
   - The rule is activated.
   - The rule is deactivated but has pending actions in the workflow queue.
3. Specify a number of days or hours before or after a date that's relevant to the record, such as the date the record was created.

   If the workflow rule is still active and valid when this time occurs, the time trigger fires the workflow action.
4. Save your time trigger.
5. In the section for the time trigger you created, click **Add Workflow Action**.
6. Select one of the options to create an action or select an existing one.
7. Click **Done**.

SEE ALSO:

Add Automated Actions to Your Workflow Rule

Considerations for Time-Dependent Actions and Time Triggers

Identify Your Salesforce Org's Default Workflow User

Select a `Default Workflow User` that you want Salesforce to display with a workflow rule when the user that triggered the rule is not active.

If your organization uses time-dependent actions in workflow rules, you must designate a default workflow user. When the user who triggered the rule isn't active, Salesforce displays the username of the default workflow user in the `Created By` field for tasks, the `Sending User` field for email, and the `Last Modified By` field for field updates. Salesforce does not display this username for outbound messages. If a problem occurs with a pending action, the default workflow user receives an email notification.

When workflow email alerts approach or exceed certain limits, Salesforce sends a warning email to the default workflow user or—if the default workflow user isn't set—to an active system administrator.

1. From Setup, enter *`Process Automation Settings`* in the `Quick Find` box, then select **Process Automation Settings**.
2. For `Default Workflow User`, select a user.
3. Save your changes.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit process automation settings:
- "Customize Application"

Associate Actions with Workflow Rules or Approval Processes

Associate actions that have already been created in your organization with a workflow rule and approval processes.

To associate existing workflow actions with a workflow rule:

1. From Setup, enter *`Workflow Rules`* in the `Quick Find` box, then select **Workflow Rules**.
2. Select the workflow rule.
3. Click **Edit** in the Workflow Actions section.
4. Click **Add Workflow Action** in either the Immediate Workflow Actions or Time-Dependent Actions section, depending on when you want the action to occur, and choose **Select Existing Action**.
5. Select the type of action to associate with the workflow rule.
6. Select the actions in the **Available Actions** box and use the right arrow to move them to the **Selected Actions** box. If necessary, select the left arrow to remove actions from the **Available Actions** box.
7. Save your changes.

To associate existing workflow actions with an approval process:

1. From Setup, enter *`Approval Processes`* in the `Quick Find` box, then select **Approval Processes**.
2. Click the name of an approval process.
3. Depending on when you want the action to occur, do one of the following:
   - Click **Add Existing** in the Initial Submission Actions, Final Approval Actions, Final Rejection Actions, or Recall Actions section.
   - Click **Show Actions** in the Approval Steps section, then click **Add Existing** in the Approval, Rejection, or Recall Actions section.

   See Add an Existing Automated Action to Your Approval Process on page 420.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To select existing actions:
- "Customize Application"

4. Select the type of action you want to associate with the approval process. The **Available Actions** box lists all existing actions of the selected type.
5. Enter the name of a specific action in the text field and click **Find**.
6. Select the actions in the **Available Actions** box that you want to associate with the approval process, and use the right arrow to move the actions to the **Selected Actions** box. If necessary, select the left arrow to remove actions from the **Available Actions** box.
7. Save your changes.

Activate Your Workflow Rule

Salesforce doesn't trigger a workflow rule until you activate it.

To activate a workflow rule, click **Activate** on the workflow rule detail page. Click **Deactivate** to prevent a rule from triggering or if you want to edit the time-dependent actions and time triggers that are associated with the rule.

You can deactivate a workflow rule at any time. However, if you deactivate a rule that has pending actions, Salesforce completes those actions as long as the record that triggered the rule is not updated.

Note:

- You can't delete a workflow rule that has pending actions in the workflow queue. Wait until pending actions are processed, or use the workflow queue to cancel the pending actions.
- You can't add time-dependent workflow actions to active workflow rules. Deactivate the workflow rule first, add the time-dependent workflow action, and reactivate the rule.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules and actions:
- "Customize Application"

SEE ALSO:

Set the Criteria for Your Workflow Rule

Workflow Considerations

Learn the intricacies of workflow rules and workflow actions before you begin working with them.

Tip: Whenever possible, automate your if/then statements with Process Builder instead of workflow rules.

- Each workflow rule applies to a single object.
- If you have workflow rules on converted leads and want to use cross-object field updates on the resulting accounts and opportunities, you must enable the lead setting `Require Validation for Converted Leads`.
- Workflow rules on custom objects are automatically deleted if the custom object is deleted.
- The order in which individual actions and types of actions are executed is not guaranteed. However, field update actions are executed first, followed by other actions.
- To create workflow rules that update case fields based on new case comments or incoming email messages, choose Case Comment or Email Message from the `Select Object` drop-down list; Email Message is only available if Email-to-Case or On-Demand Email-to-Case is enabled. You can only create email message workflow rules for field updates, and case comment workflow rules

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

for field updates, email alerts, and outbound messages. For example, you can create a workflow rule so that an email marked as `Is Incoming` changes its case's `Status` from Closed to New.

- Changes you make to records while using Connect Offline are evaluated by workflow rules when you synchronize.
- Salesforce processes rules in the following order:
  1. Validation rules
  2. Assignment rules
  3. Auto-response rules
  4. Workflow rules (with immediate actions)
  5. Escalation rules
- If a lookup field references a record that has been deleted, Salesforce clears the value of the lookup field by default. Alternatively, you can choose to prevent records from being deleted if they're in a lookup relationship.
- If you create workflow rules to replace any Apex triggers, make sure you delete those Apex triggers when you activate the equivalent workflow rules. Otherwise, both Apex triggers and workflow rules will fire and cause unexpected results, such as overwritten field updates or redundant email messages.

When Do Workflow Rules Get Triggered?

- Workflow rules can be triggered any time a record is saved or created, depending on your rule criteria. However, rules created after saving records aren't triggered by those records retroactively.
- Workflow rules are triggered when a standard or custom object in a master-detail relationship is re-parented, even if the object's evaluation criteria is set to `Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria`.
- Saving or creating records can trigger more than one rule.
- Workflow rules only trigger on converted leads if validation and triggers for lead convert are enabled in your organization.
- Workflow rules trigger automatically and are invisible to the user. Alternatively, approval processes allow users to submit records for approval.
- If your organization uses multiple languages, enter filter values in your individual language. You can add up to 25 filter criteria, of up to 255 characters each.

  When you use picklists to specify filter criteria, the selected values are stored in the organization's default language. If you edit or clone existing filter criteria, first set the `Default Language` on the Company Information page to the same language that was used to set the original filter criteria. Otherwise, the filter criteria may not be evaluated as expected.
- If you use record types in your workflow rule criteria whose labels have been translated using the translation workbench, the translated label value won't trigger the workflow rule. Workflow criteria evaluate the master label value and ignore the translated value. To avoid this problem, set the workflow criteria to evaluate the master record type label value by entering it manually in the `Value` field.
- Workflow rules can't be triggered by campaign statistic fields, including individual campaign statistics and campaign hierarchy statistics.
- A workflow rule isn't triggered if its condition references a field that doesn't have a value. For example, if a User-based workflow rule checks "Role not equal to CEO", the rule isn't triggered for a user without an assigned role. Instead of conditions, use a formula to check that that the field is either null or set to something other than "CEO":

  ```
  UserRoleId == null || UserRole.Name != "CEO"
  ```
- The following actions don't trigger workflow rules.

- Mass replacing picklist values
- Mass updating address fields
- Mass updating divisions
- Changing the territory assignments of accounts and opportunities
- Converting leads to person accounts
- Deactivating Self-Service Portal, Customer Portal, or Partner Portal users
- Converting state and country data from the State and Country Picklists page in Setup
- Making changes to state and country picklists using AddressSettings in the Metadata API

Workflow Rule Limitations

- You can't package workflow rules with time triggers.
- You can't create outbound messages for workflow rules on junction objects.

Tip: Use the Developer Console to debug workflow rules. The Developer Console lets you view debug log details and information about workflow rules and actions, such as the name of the user who triggered the workflow rule and the name and ID of the record being evaluated.

IN THIS SECTION:

Workflow for the User Object (Beta)
You can create workflow rules and actions for the User object. You can, for example, send welcome emails to new employees or sync user data with a third-party service using outbound message actions.

Considerations for Time-Dependent Actions and Time Triggers

SEE ALSO:

Set the Criteria for Your Workflow Rule

Workflow for the User Object (Beta)

You can create workflow rules and actions for the User object. You can, for example, send welcome emails to new employees or sync user data with a third-party service using outbound message actions.

Note: This release contains a beta version of workflow on the User object that is production quality but has known limitations. To provide feedback and suggestions, go to IdeaExchange.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example Use Cases

For the User object, you can set up workflow rules to:

- Send welcome email messages with training resources to newly created users by using email alert actions.
- Send emails when users change roles or are deactivated by using email alert actions.
- Deactivate temporary employees after a specified period by using field update actions.
- Sync user data with third-party systems by using outbound messages actions.

Merge Field Types for the User Object

To use merge fields from user records in email templates, select from the following merge field types:

- User Fields—Use these merge fields to represent the sending user. Merge fields named `{!User.field_name}` return values from the user record of the person who created or updated the record that triggered the workflow rule.
- Workflow Target User Fields—Use these merge fields only in email templates for workflow rules on the User object. Merge fields named `{!Target_User.field_name}` return values from the user record that was created or updated to trigger the workflow rule.

Beta Limitations for Workflow on the User Object

Understand these limitations before you create workflow rules or workflow actions for the User object.

- Tasks are not supported as workflow actions for the User object.
- When setting the workflow rule criteria, you can't select `Current User` fields using the picklists. You can, however, use a formula to set the rule criteria and include fields from the current user. In the formula editor, click **Insert Field**, select `$User`, select the field, and click **Insert**.
- Remember that custom validation rules run *before* workflow rules are executed. Refer to "Triggers and Order of Execution" in the *Apex Developer Guide*.

SEE ALSO:

Workflow Considerations

Considerations for Time-Dependent Actions and Time Triggers

When creating time-dependent actions and time triggers for workflow rules, consider the following.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Defining Time Triggers

- When defining a time trigger, use standard and custom date and date/time fields defined for the object. Specify time using days and hours. The valid range is 0 to 999 days or hours.
- You can modify existing time triggers by adding or removing actions.

  Note: Removing all the actions from a time trigger does not remove the trigger. Empty triggers are still queued and count against your hourly workflow time trigger limit. To remove scheduled time triggers, delete them from the workflow queue.

Time Trigger Processing

- Time-dependent actions aren't executed independently. They're grouped into a single batch that starts executing within one hour after the first action enters the batch.
- Apex triggers that fire as a result of time-dependent actions may get executed in a single batch or independently. Follow these best practices:
  - In case they fire independently–Ensure that your Apex logic is scoped for a single scheduled action. For example, don't use Apex static variables to communicate state across Apex code triggered by different scheduled actions.
  - In case they fire in a single batch–Be aware of how the combination of your time-dependent actions and Apex triggers impact your Apex governor limits.

- Salesforce evaluates time-based workflow on the organization's time zone, not the user's. Users in different time zones might see differences in behavior.
- Salesforce doesn't necessarily execute time triggers in the order they appear on the workflow rule detail page. Workflow rules list time triggers that use the `Before` field first, followed by time triggers that use the `After` field.
- Salesforce doesn't display time-dependent action controls on the workflow rule edit page if you set the workflow rule evaluation criteria to `Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria`.
- If you change a date field that is referenced by an unfired time trigger in a workflow rule that has been evaluated, Salesforce recalculates the unfired time triggers associated with the rule. For example, if a workflow rule is scheduled to alert the opportunity owner 7 days before the opportunity close date and the close date is set to 2/20/2011, Salesforce sends the alert on 2/13/2011. If the close date is updated to 2/10/2011 and the time trigger hasn't fired, Salesforce reschedules the alert for 2/3/2011. If Salesforce recalculates the time triggers to a date in the past, Salesforce triggers the associated actions shortly after you save the record.
- If a workflow rule has a time trigger set for a time in the past, Salesforce queues the associated time-dependent actions to start executing within one hour. For example, if a workflow rule on opportunities is configured to update a field 7 days before the close date, and you create an opportunity record with the close date set to today, Salesforce starts to process the field update within an hour after you create the opportunity.
- Time-dependent actions remain in the workflow queue only as long as the workflow rule criteria are still valid. If a record no longer matches the rule criteria, Salesforce removes the time-dependent actions queued for that record.

  For example, an opportunity workflow rule can specify:

  - A criteria set to "Opportunity: Status not equals to Closed Won, Closed Lost"
  - An associated time-dependent action with a time trigger set to 7 days before the opportunity close date

  If a record that matches the criteria is created on July 1 and the `Close Date` is set to July 30, the time-dependent action is scheduled for July 23. However, if the opportunity is set to "Closed Won" or "Closed Lost" before July 23, the time-dependent action is removed from the queue.

- Salesforce ignores time triggers that reference null fields.
- Time-dependent actions can automatically be queued again if the record is updated and the evaluation criteria is set to `Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria`. Using the previous example, if the opportunity status is changed from Closed Lost to Prospecting and the workflow rule evaluation criteria is `Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria`, Salesforce reevaluates the time triggers and adds the appropriate actions to the workflow queue.
- Deleting a record that has pending actions removes the pending actions from the workflow queue. You can't restore the actions, even if you undelete the record.
- If the evaluation criteria is set to `Evaluate the rule when a record is: created`, the workflow rule evaluates its time triggers only once. If the record that fired the rule changes to no longer meet the evaluation criteria, Salesforce removes the pending actions from the queue and never reapplies the rule to the record.
- You can deactivate a workflow rule at any time. If the rule has pending actions in the workflow queue, editing the record that triggered the rule removes the pending actions from the queue. If you don't edit the record, the pending actions are processed even though the rule has been deactivated.
- Time-dependent actions aren't executed for a reevaluated workflow rule in the following situations:
  - The reevaluated workflow rule's immediate actions cause the record to no longer meet the workflow rule criteria.
  - An Apex `after` trigger that is executed as a result of a workflow or approvals action causes the record to no longer meet the workflow rule criteria.

- Configuring a task's `Due Date` to "Rule Trigger Date" sets time triggers and workflow task due dates based on the date that the workflow time trigger's action is executed. For example, if the task due date is "Rule Trigger Date plus 10 days" and the time trigger is executed on January 1, Salesforce sets the task due date to January 11.
- You can add a new active workflow rule with time triggers in a change set and deploy it. You can only change time triggers on a workflow rule in a change set if it's inactive. The rule must be activated in the destination organization manually or through another change set that only activates workflow rules and makes no time trigger changes.

  For example, let's say you have an inactive workflow rule in your destination organization, and your change set contains an active workflow rule with the same name and new or different time triggers. The deployment fails because it activates the workflow rule first and then tries to add or remove the time triggers.

  Note: You must add time-dependent actions manually when including a workflow rule in a change set. The **View/Add Dependencies** function doesn't detect time-dependent actions.

Using Time-Dependent Workflow with Leads

- You can't convert a lead that has pending actions.
- If Validation and Triggers from Lead Convert is enabled, existing time-based workflow actions on leads aren't triggered during lead conversion.
- If a campaign member based on a lead is converted before the completion of the time-based workflow actions associated with it, Salesforce still performs the time-based workflow actions.

Limitations

- Time triggers don't support minutes or seconds.
- Time triggers can't reference the following:
  - `DATE` or `DATETIME` fields containing automatically derived functions, such as `TODAY` or `NOW`.
  - Formula fields that include related-object merge fields.
- Salesforce limits the number of time triggers an organization can execute per hour. If an organization exceeds the limits for its Edition, Salesforce defers the execution of the additional time triggers to the next hour. For example, if an Unlimited Edition organization has 1,200 time triggers scheduled to execute between 4:00 PM and 5:00 PM, Salesforce processes 1,000 time triggers between 4:00 PM and 5:00 PM and the remaining 200 time triggers between 5:00 PM and 6:00 PM.
- You can't archive a product or price book that has pending actions.
- If time-based workflow actions exist in the queue, you can't add or remove time triggers or edit trigger dates without deleting the actions first. Because the deleted records can't be restored, carefully consider the implications of editing the workflow rules before you proceed. If you decide to edit the workflow rules, deactivate the workflow that you want to edit, edit the rules as needed, and then save your changes. For information about finding and deleting time-based workflow actions in the queue, see Monitor Pending Workflow Actions on page 405.

  You also can't add or remove time triggers if:

  - The workflow rule is active.
  - The workflow rule is deactivated, but has pending actions in the queue.
  - The workflow rule evaluation criteria is set to `Evaluate the rule when a record is: created, and every time it's edited`.

- The workflow rule is included in a package.

SEE ALSO:

Add Automated Actions to Your Workflow Rule

Identify Your Salesforce Org's Default Workflow User

Workflow Rule Examples

Looking for ideas on how workflow rules can help streamline your business? Check out these examples.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Business Processes

Follow Up Before Contract Expires

Follow Up when Platinum Contract Case Closes

Assign Credit Check for New Customer

Notify Account Owner About New, High-Priority Cases

Set a Default Entitlement for Each New Case

Update Shipment Status if Shipment is Delayed

Automatically Activate New Users

Cross-Object Processes

Notify Sales VP About Cases Filed for Top Accounts

Set Default Opportunity Name

Set Target Resolution Date for Cases

Update Application Record when Candidate Accepts Job

Deal Management

Track Closed Opportunities

Override Default Opportunity Close Date

Report Lost Opportunities

Report Unassigned Leads

Send Alert if Quote Line Item Discount Exceeds 40%

Notifications

Notify Key People About Account Owner Changes

Set Reminder for Contact Birthday

Set Reminder for High-Value Opportunity Close Date

Notify Account Owner of Updates by Others

Follow Up Before a Contract Expires

Object	Contract
Description	Email a reminder to the renewal manager 20 days before a contract's end date.

Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(Contract: Status equals Activated)`
Immediate Actions	None
Time-Dependent Actions	20 Days Before Contract: End Date—`Email Alert:` Email a reminder to the renewal manager to confirm whether the client wants an extension.

Follow Up When a Platinum Contract Case Closes

This example assumes that a `Contract Type` custom picklist is used to identify the contract level on cases and that the picklist contains the Platinum value.

Object	Case
Description	If the customer has a platinum contract agreement, email a feedback request to the case contact 7 days after a high-priority case has been closed.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.

Rule Criteria (Filter)	Run this rule if the following criteria are met.
	`(Case: Priority equals High) and`
	`(Case: Closed equals True) and`
	`(Case: Contract Type equals Platinum)`

Immediate Actions	None
Time-Dependent Actions	7 Days After Case: Date/Time Closed—`Email Alert:` Email a feedback request to the case contact.

Assign Credit Check for a New Customer

This example assumes that a `New Customer` custom field is on opportunities.

Object	Opportunity
Description	Assign the Accounts Receivable (AR) department a task to check the credit of a potential customer 15 days before the opportunity close date if the amount is greater than $50,000.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.

Rule Criteria (Filter)	Run this rule if the following criteria are met.
	`(Opportunity: Amount greater than 50000) and`
	`(Opportunity: Closed equals False) and`
	`(Opportunity: New Customer equals True)`

Immediate Actions	None
Time-Dependent Actions	15 Days Before Opportunity: Close Date—`Task:` Create a task for users in the Accounts Receivable role to run a credit check.

Notify Account Owner About New, High-Priority Cases

This example assumes that a Service Level Agreement custom picklist called SLA identifies the agreement level on accounts and contains the Platinum value.

Object	Case
Description	Notify the account owner when a high-priority case is created for accounts with a platinum SLA.
Evaluation Criteria	Evaluate the rule when a record is: `created`.

Rule Criteria (Filter)	Run this rule if the following criteria are met.
	`(Case: Priority equals High) and`
	`(Account: SLA equals Platinum)`

Immediate Actions	`Email Alert:` Email the details of the high-priority case to the account owner.
Time-Dependent Actions	None

Set a Default Entitlement for Each New Case

This example assumes that an active, autolaunched flow looks up the relevant entitlement based on the account, asset, or contact associated with the new case and updates the case with the entitlement name.

The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Object	Case
Description	Set a default entitlement on each new case.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(Case: Status not equal to Closed)`
Immediate Actions	`Flow Trigger:` Look up and assign the relevant entitlement to the case. Pass the account, asset, or contact associated with the new case into the relevant flow variable to enable the entitlement lookup. Pass the case ID into the relevant flow variable to enable the case update.
Time-Dependent Actions	None.

Update Shipment Status If Shipment Is Delayed

Object	Shipment
Description	Update the `Shipment Status` field to Delayed if a shipment has exceeded the expected delivery date and hasn't reached the customer.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(Shipment: Status not equal to Delivered)`
Immediate Actions	None
Time-Dependent Actions	1 day after Shipment: Expected Delivery Date—`Field Update`: Change `Shipment Status` field to Delayed on Shipment record.

Automatically Activate New Users

Object	User
Description	Make sure that each new user is active so that the user can log in to Salesforce.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(User: Active equals False)`
Immediate Actions	`Field Update`: Set `Active` to True.
Time-Dependent Actions	None.

Notify Sales VP About Cases Filed for Top Accounts

This workflow rule is for sales VP who want to know about cases filed for top accounts. Top accounts are determined by size and revenue.

Object	Case
Description	Notify sales VP about cases filed for top accounts.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following criteria are met. `AND(Account.AnnualRevenue > 500000, Account.NumberOfEmployees > 5000)`
Immediate Actions	`Email Alert:` Notify VP about cases for large accounts.
Time-Dependent Actions	None

Set Default Opportunity Name

The opportunity naming convention for some companies is *`Account Name: Opportunity Name`*. To automate the default name of each opportunity in your org, create the following workflow rule.

Object	Opportunity
Description	Enforce opportunity naming convention.
Evaluation Criteria	Evaluate the rule when a record is: `created, and every time it's edited`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `NOT(CONTAINS( Name, Account.Name ))`
Immediate Actions	`Field Update`: Set opportunity name to the following formula. `Account.Name & ": " & Name`
Time-Dependent Actions	None

Set Target Resolution Date for Cases

This example sets a case resolution date based on the value of a field on the associated account. It uses a custom picklist field on accounts called `Support Level`, which has three values: Basic, Standard, and Premium. It also has a custom date field on cases called `Target Resolution Date`.

Use the following three workflow rule examples to set the target resolution date of a case based on the support level for the related account.

Set Resolution Date for Basic Support

Object	Case
Description	Set the case target resolution date for accounts that have basic support level to 30 days from today.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following formula is true. `ISPICKVAL(Account.Support_Level__c , "Basic")`
Immediate Actions	`Field Update`: Set the `Target Resolution Date` to Today() + 30.
Time-Dependent Actions	None

Set Resolution Date for Standard Support

Object	Case
Description	Set the case target resolution date for accounts that have standard support level to 14 days from today.
Evaluation Criteria	Evaluate the rule when a record is: `created`.

Rule Criteria (Filter)	Run this rule if the following formula is true. `ISPICKVAL(Account.Support_Level__c , "Standard")`
Immediate Actions	`Field Update`: Set the `Target Resolution Date` to Today() + 14.
Time-Dependent Actions	None

Set Resolution Date for Premium Support

Object	Case
Description	Set the case target resolution date for accounts that have premium support level to 5 days from today.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following formula is true. `ISPICKVAL(Account.Support_Level__c , "Premium")`
Immediate Actions	`Field Update`: Set the `Target Resolution Date` to Today() + 5.
Time-Dependent Actions	None

Update Application Record When Candidate Accepts Job

This workflow rule closes the Application record when a candidate accepts the job. Cross-object field updates to the master record are supported between custom objects in a master-detail relationship.

Object	Candidate
Description	Change the `Application Status` field to Closed for the custom Application object when the `Candidate Status` field for the custom Candidate object changes to Accepted.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(Candidate: Status equals Accepted)`
Immediate Actions	`Field Update:` Change the `Application Status` field to Closed on parent Application record.
Time-Dependent Actions	None

Track Closed Opportunities

This example assumes that a Closed Opportunities record type provides additional information to certain profiles.

Note: For information on record types, see Record Types on page 57.

Object	Opportunity
Description	Change the record type of closed-won opportunities.
Evaluation Criteria	Evaluate the rule when a record is: `created, and every time it's edited`.

Rule Criteria (Filter)	Run this rule if the following criteria are met.
	`(Opportunity: Closed equals True) and`
	`(Opportunity: Stage equals Closed Won)`

Immediate Actions	`Field Update:` Set the record type to Closed Opportunities.
Time-Dependent Actions	None

Override the Default Opportunity Close Date

Object	Opportunity
Description	Override the default close date from the close of the quarter to 6 months after the opportunity is created.
Evaluation Criteria	Evaluate the rule when a record is: `created`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `(Opportunity: Closed equals False)`
Immediate Actions	`Field Update:` Use the following formula to set the opportunity close date to 6 months after the creation date. `DATE( YEAR(TODAY()) , (MONTH(TODAY()) + 6), DAY(TODAY()))`
Time-Dependent Actions	None

Report Lost Opportunities

Object	Opportunity
Description	Notify the VP of sales when a deal is lost if the stage was Proposal/Price Quote and the amount was greater than $1 million.
Evaluation Criteria	Evaluate the rule when a record is: `created, and every time it's edited`.
Rule Criteria (Filter)	Run this rule if the following formula is true. `AND( ISCHANGED(StageName), ISPICKVAL(PRIORVALUE(StageName) , "Proposal/Price Quote"), ISPICKVAL(StageName,"Closed Lost"), (Amount >1000000))`
Immediate Actions	`Email Alert:` Notify the VP of sales role that the deal was lost.
Time-Dependent Actions	None

Report Unassigned Leads

This example assumes that all unassigned leads are placed in an unassigned leads queue by a leads assignment rule.

Object	Lead
Description	Ensure that unassigned leads are tracked in a timely manner by notifying the manager if a lead is not accepted in 2 days.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `Lead Owner equals Unassigned Lead Queue`
Immediate Actions	None
Time-Dependent Actions	2 Days After Lead: Last Modified Date—`Email Alert:` Notify the manager role that the queue has unassigned leads that are older than 2 days.

Send Alert If Quote Line Item Discount Exceeds 40%

Object	Quote Line Item
Description	Ensure that an email alert is sent if a sales rep applies a quote line item discount that exceeds 40%.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.
Rule Criteria (Filter)	Run this rule if the following criteria is met. `Quote Line Item: Discount is greater than 40`
Immediate Actions	`Email Alert:` Notify the manager role that the quote line item discount exceeds 40%.
Time-Dependent Actions	None

Notify Key People About Account Owner Changes

Object	Account
Description	Notify key people in the sales department when the owner of an account changes if the account's annual revenue is greater than $1 million.
Evaluation Criteria	Evaluate the rule when a record is: `created, and every time it's edited`.
Rule Criteria (Filter)	Run this rule if the following formula is true. `AND( ISCHANGED(OwnerId), AnnualRevenue > 1000000 )`
Immediate Actions	`Email Alert:` Notify the person in the sales operations role of the change in account ownership.

Time-Dependent Actions	None

Set Reminder for Contact Birthday

This example assumes that a `Next Birthday` custom formula field uses the following formula to calculate the date of the contact's next birthday on contact records.

```
IF(MONTH(Birthdate) > MONTH(TODAY()),DATE(YEAR(TODAY()),MONTH(Birthdate),DAY(Birthdate)),
IF(MONTH(Birthdate) < MONTH(TODAY()),DATE(YEAR(TODAY())+1,MONTH(Birthdate),DAY(Birthdate)),
IF(DAY(Birthdate) >= (DAY(TODAY())),DATE(YEAR(TODAY()),MONTH(Birthdate),DAY(Birthdate)),
DATE(YEAR(TODAY())+1,MONTH(Birthdate),DAY(Birthdate)))))
```

Object	Contact
Description	Send an email to the contact 2 days before the contact's birthday.
Evaluation Criteria	Evaluate the rule when a record is: `created`.

Rule Criteria (Filter)	Run this rule if the following formula is true.
	`(Contact: Birthdate not equal to null) and`
	`(Contact: Email not equal to null)`

Immediate Actions	None
Time-Dependent Actions	2 Days Before Contact: Next Birthday—`Email Alert:` Send a birthday greeting to the contact's email address.

Set Reminder for High-Value Opportunity Close Date

Object	Opportunity
Description	Remind the opportunity owner and senior management when the close date is approaching for an opportunity that has an amount greater than $100,000. Create a follow-up task for the opportunity owner if the deal is still open when the close date passes.
Evaluation Criteria	Evaluate the rule when a record is: `created, and any time it's edited to subsequently meet criteria`.

Rule Criteria (Filter)	Run this rule if the following criteria are met.
	`(Opportunity: Amount greater than 100000) and`
	`(Opportunity: Closed equals False)`

Immediate Actions	None
Time-Dependent Actions	• 30 Days Before Opportunity: Close Date—`Email Alert:` Notify the opportunity owner that 30 days remain. • 15 Days Before Opportunity: Close Date—`Email Alert:` Notify the opportunity owner that 15 days remain.

- 5 Days After Opportunity: Close Date—`Task:` Create a follow-up task for the opportunity owner to update the deal. `Email Alert:` Notify senior management to involve executives.

Notify Account Owner of Updates by Others

Object	Account
Description	Notify the account owner when someone else updates the account if the account's annual revenue is greater than $1 million.
Evaluation Criteria	Evaluate the rule when a record is: `created, and every time it's edited`.
Rule Criteria (Filter)	Run this rule if the following formula is true. `AND( (LastModifiedById <> OwnerId), (AnnualRevenue > 1000000) )`
Immediate Actions	`Email Alert:` Notify the account owner that someone else has updated the account.
Time-Dependent Actions	None

SEE ALSO:

Workflow

Set the Criteria for Your Workflow Rule

Monitor Pending Workflow Actions

When a workflow rule that has time-dependent actions is triggered, use the workflow queue to view pending actions and cancel them if necessary.

1. From Setup, enter *`Time-Based Workflow`* in the `Quick Find` box, then select **Time-Based Workflow**.
2. Click **Search** to view all pending actions for any active workflow rules, or set the filter criteria and click **Search** to view only the pending actions that match the criteria. The filter options are:
   - **Workflow Rule Name**: The name of the workflow rule.
   - **Object**: The object that triggered the workflow rule. Enter the object name in the singular form.
   - **Scheduled Date**: The date the pending actions are scheduled to occur.
   - **Create Date**: The date the record that triggered the workflow was created.
   - **Created By**: The user who created the record that triggered the workflow rule.
   - **Record Name**: The name of the record that triggered the workflow rule.

   The filter is not case-sensitive.

To cancel pending actions:

1. Select the box next to the pending actions you want to cancel.
2. Click **Delete**.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Workflow tasks and email alerts are not available in **Database.com**

USER PERMISSIONS

To manage the workflow queue:

- "Modify All Data"

Workflow Terminology

The following terms are used when describing workflow features and functionality.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Flow triggers are not available in Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Email alerts and outbound messages available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Workflow Rule

A workflow rule sets workflow actions into motion when its designated conditions are met. You can configure workflow actions to execute immediately when a record meets the conditions in your workflow rule, or set time triggers that execute the workflow actions on a specific day. If a workflow action hasn't executed yet, you can view and modify it in the workflow queue.

Workflow Action

A workflow action, such as an email alert, field update, outbound message, or task, fires when the conditions of a workflow rule are met.

Email Alert

Email alerts are actions that send emails, using a specified email template, to specified recipients. Workflow alerts can be sent to any user or contact, as long as they have a valid email address.

Field Update

A field update is an action that automatically updates a field with a new value.

Flow

A *flow* is an application that can execute logic, interact with the Salesforce database, call Apex classes, and collect data from users. You can build flows by using the Cloud Flow Designer.

Flow Trigger

A *flow trigger* is a workflow action that launches a flow. With flow triggers, you can automate complex business processes—create flows to perform logic, and have events trigger the flows via workflow rules—without writing code.

The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Outbound Message

An outbound message sends information to a designated endpoint, like an external service. Outbound messages are configured from Setup. You must configure the external endpoint and create a listener for the messages using the SOAP API.

Approvals

It's likely that you're familiar with process automation in the form of workflow rules. Approvals take automation one step further, letting you specify a sequence of steps that are required to approve a record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including who to request approval from and what to do at each point of the process.

Example: Your org has a three-tier process for approving expenses. This approval process automatically assigns each request to right person in your org, based on the amount requested.

If an expense record is submitted for approval, lock the record so that users cannot edit it and change the status to Submitted.

If the amount is $50 or less, approve the request. If the amount is greater than $50, send an approval request to the direct manager. If the amount is greater than $5,000 and the first approval request is approved, send an approval request to the vice president.

If all approval requests are approved, change the status to Approved and unlock the record. If any approval requests are rejected, change the status to Rejected and unlock the record.

IN THIS SECTION:

Set Up an Approval Process

If Approvals is the right automation tool for your business process, follow these high-level steps to create one for your org.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval processes:
- "Customize Application"

IN THIS SECTION:

1. Prepare to Create an Approval Process
   Plan each approval process carefully to ensure a successful implementation.
2. Choose the Right Wizard to Create an Approval Process
   Before you create an approval process, determine which wizard is best for your needs.
3. Add an Approval Step to an Approval Process
   Approval steps define the chain of approval for a particular approval process. Each step determines which records can advance to that step, who to assign approval requests to, and whether to let each approver's delegate respond to the requests. The first step specifies what to do if a record doesn't advance to that step. Later steps specify what happens if an approver rejects the request.
4. Add Automated Actions to an Approval Process
   You can associate actions to approval steps, initial submission, final approval, final rejection, or recall. Approval processes support four automated actions.
5. Review an Approval Process
   Generate a graphical representation of your approval process before you activate it. Use the Process Visualizer to gain buy-in from step owners and reinforce your company's policies by documenting the decisions you reached when the approval process was designed.

6. Activate an Approval Process
   After you've created at least one step for the approval process, activate the process.

SEE ALSO:

- Approval Process Terminology
- Sample Approval Processes
- Prepare Your Org for Approvals

Prepare to Create an Approval Process

Plan each approval process carefully to ensure a successful implementation.

Review the following checklist before creating your approval process.

- Prepare an approval request email template.
- Prepare an approval request post template.
- Determine the approval request sender.
- Determine the assigned approver.
- Determine the delegated approver.
- Decide if your approval process needs a filter.
- Design initial submission actions.
- Decide if users can approve requests from a wireless device.
- Determine if users can edit records that are awaiting approval.
- Decide if records should be auto-approved or rejected.
- Determine how many levels your process has.
- Determine the actions when an approval request is approved or rejected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Which email template do you want to use for approval requests?

The email template you specify on an approval process is used when notifying users that an approval request is assigned to them. You can use the Salesforce default email template or create your own template. Include the appropriate approval process merge fields to link directly to the approval request.

Does your org have email approval response enabled? If so, the default email template includes instructions for replying to an approval request by typing `approve`, `approved`, `yes`, `reject`, `rejected`, or `no` in the first line of the email body and optionally adding comments in the second line.

Which Chatter post template do you want to use for approval requests?

If your org has Approvals in Chatter enabled, specify an approval post template to use when notifying a user via Chatter about an assigned approval request. You can use the Salesforce default post template or create your own.

Who is the sender of approval requests?

Approval request notifications are sent from the user who submitted the record for approval. When you configure an email alert, you can add a different return email address for these notifications. You can choose the email address of the default workflow user or a previously configured and verified org-wide address. Determine which email address to use.

Who can approve requests?

Any of the following can approve or reject a request.

- A user or queue that the approval request submitter chooses.
- A queue specified by the administrator.

- A user listed in the `Manager` standard field on the submitter's user detail page.
- A user listed in a custom hierarchy field on the submitter's user detail page.
- Any combination of users and related users (users listed in a standard or custom field on the submitted record) specified by the administrator.

Do you want approval requests delegated to another user for approval?

An approver can designate a delegate to approve requests, but you can disable this option. To assign delegates, for each user, populate the `Delegated Approver` field on the user's detail page.

Which records are included in this process?

Determine what attributes a record must have to be included in your approval process. If necessary, create the custom fields to store this information so that you can use it in your filter criteria. For example, if you want to include expense records from your headquarters office only, create a custom picklist field called `Office Location` that has two options: "HQ" and "Field." Then, you would specify in your filter criteria that records must have "HQ" in the `Office Location` field to be included in the approval process.

What happens when a record is first submitted for approval?

When users submit a record for approval, Salesforce automatically locks the record so that other users cannot change it while it is awaiting approval. You can still add campaign members to campaigns locked for approval.

Decide if you want other workflow actions to happen when a record is first submitted, such as email alerts, tasks, field updates, and outbound messages. These actions become your initial submission actions.

Can users approve requests from a mobile device?

Determine if you want to require users to log in to Salesforce to approve requests. You can also set up your approval process to allow users to approve requests remotely using a mobile browser.

Who can edit records that are awaiting approval?

Records submitted for approval are locked. Users with the "Modify All" object-level permission for the given object or the "Modify All Data" permission can always unlock a record and edit it. You can also specify that the currently assigned approver can edit the record. You can still add campaign members to campaigns locked for approval.

Can records be automatically approved, rejected, or skipped based on certain criteria?

You can set entry criteria for each step of your process and configure Salesforce to approve, reject, or skip the process if a record doesn't meet the criteria. For example, all expenses submitted with an `Amount` less than $15 are automatically approved.

How many people need to approve these requests?

An approval process can have several layers of approvals. Determine how many users need to approve requests and in what order.

What should happen when a request is approved or rejected?

When a request is recalled, approved, or rejected, Salesforce can perform up to 10 instances of each of the following types of actions—up to 40 actions total. You can also configure up to 40 actions to occur when a record has received all necessary approvals or is rejected.

SEE ALSO:

Set Up an Approval Process
Considerations for Approvals
Sample Approval Processes

Choose the Right Wizard to Create an Approval Process

Before you create an approval process, determine which wizard is best for your needs.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Create an Approval Process with the Jump Start Wizard
For approval processes that use a single step, use the jump start wizard. This wizard chooses some default options for you.

Default Selections for the Approval Process Jump Start Wizard
To make it easier for you to get started with a simple approval process, the jump start wizard automatically chooses some default options for you.

Create an Approval Process with the Standard Wizard
When your approval process is more complex and you want to define specific steps, use the standard wizard.

SEE ALSO:

Set Up an Approval Process

Create an Approval Process with the Jump Start Wizard

For approval processes that use a single step, use the jump start wizard. This wizard chooses some default options for you.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval processes:
- "Customize Application"

1. From Setup, enter *Approval Processes* in the Quick Find box, then select **Approval Processes**.
2. Select an object.
3. Select **Create New Approval Process** > **Use Jump Start Wizard**.
4. Configure the approval process by following the wizard.
   a. Default Selections for the Approval Process Jump Start Wizard
   b. Choose Approval Request Notification Templates
   c. Design the Approval Request Page
   d. Control Which Records Apply to an Approval Process
   e. Identify Assigned Approvers for an Approval Step

SEE ALSO:

Default Selections for the Approval Process Jump Start Wizard
Considerations for Configuring Approvals
Considerations for Setting Approvers
Set Up an Approval Process
Choose the Right Wizard to Create an Approval Process

Default Selections for the Approval Process Jump Start Wizard

To make it easier for you to get started with a simple approval process, the jump start wizard automatically chooses some default options for you.

After creating an approval process using the jump start wizard, you can modify these default options and add more steps from the approval process detail page. Exception: you can't modify the Record Lock action on the Initial Submission Actions list.

- To edit records awaiting approval in the approval process, users must have the "Modify All" permission for the given object or the "Modify All Data" permission.
- The page layout for the approval request includes the record name (or number), owner, date created, and approval history.
- The security settings require approvers to log in to Salesforce to view the approval page.
- Only the owner of the record can submit the record for approval.
- Records are locked when submitted for approval.
- Records remain locked until approved or rejected.
- Rejected records are unlocked.
- Only admins can recall a record after it's submitted.
- There are no auto-approve or auto-reject actions.
- No email notification is sent upon approval or rejection.
- No field values are automatically updated during the approval process.
- An approver cannot automatically delegate another user to approve the approval requests.
- The **Allow submitters to recall approval requests** option is not selected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Create an Approval Process with the Jump Start Wizard

Choose the Right Wizard to Create an Approval Process

Create an Approval Process with the Standard Wizard

When your approval process is more complex and you want to define specific steps, use the standard wizard.

From Setup, enter *Approval Processes* in the Quick Find box, then select **Approval Processes**.

Select an object, and then select **Create New Approval Process** > **Use Standard Setup Wizard**. Configure the approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval processes:

- "Customize Application"

IN THIS SECTION:

1. Control Which Records Apply to an Approval Process
   Narrow down the list of records that can be part of the approval process by specifying criteria. You can either use filters or write a formula.
2. Choose Approval Request Notification Templates
   When an approval process assigns an approval request to a user, Salesforce sends the user an approval request email. If Approvals in Chatter is enabled, Salesforce also posts the approval request to Chatter. Choose templates for each of these notifications.

3. Choose an Automated Approver Throughout an Approval Process
   Associate a hierarchy field—such as the user's manager—with an approval process. When selected, the field is available as an assigned approver option for approval steps. You can always select a hierarchy field here but not use it for any approval steps.
4. Specify Who Can Edit Locked Records
   When a record is submitted for approval, it's locked to prevent users from editing it during the approval process. Use the record editability properties to determine who can edit records that are locked in this approval process.
5. Design the Approval Request Page
   The approval page is where an approver responds to an approval request. Customize which fields appear on that page and in which order. This page is used only for this approval process.
6. Specify Who Can Submit Records to an Approval Process
   Only specified individuals or roles can submit a record for approval. You can also let submitters recall an approval request.

SEE ALSO:

Set Up an Approval Process

Considerations for Approvals

Control Which Records Apply to an Approval Process

Narrow down the list of records that can be part of the approval process by specifying criteria. You can either use filters or write a formula.

If you want all records to pass through the approval process, click **Next**. If only certain types of records are considered, use one of the following options.

Option	To enter the approval process...
criteria are met	The record must meet the filter criteria.
formula evaluates to true	The formula must return `True`. Some functions aren't available in approval process formulas. For information on which functions you can use in approval process formulas, see Formula Operators and Functions.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: This filter lets an expense report enter this approval process only if the employee who submitted the report is located at headquarters.

```
Current User: Office Location Equals Headquarters
```

This formula lets a record enter this approval process only if its discount approval cutoff date is less than 30 days away.

```
(Discount_Approval_CutoffDate__c < (CloseDate - 30)
```

SEE ALSO:

Considerations for Configuring Approvals

Formula Operators and Functions

Choose Approval Request Notification Templates

When an approval process assigns an approval request to a user, Salesforce sends the user an approval request email. If Approvals in Chatter is enabled, Salesforce also posts the approval request to Chatter. Choose templates for each of these notifications.

These fields are available from both the jump start and standard wizards.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Approval Assignment Email Template	Choose a custom email template to use when notifying approvers that an approval request is assigned to them. Or leave blank to use the default email template. The approval process uses the same template for every assignment email—no matter which approval step it's for.
Approval Post Template	Available only when Approvals in Chatter is enabled. Choose an approval post template to use when notifying approvers via a post in their Chatter feed. Leave blank to use the default post template for this object or, if there isn't one, the system default template.

Note: If email approval response is enabled, be sure that the email template you use describes how to correctly use both response options: clicking the link and replying by email. If the user doesn't respond correctly (for example, if the user misspells `approve` or types it on the wrong line), Salesforce doesn't register the user's response.

SEE ALSO:

- Chatter Post Templates for Approval Requests
- Merge Fields for Approvals

Choose an Automated Approver Throughout an Approval Process

Associate a hierarchy field—such as the user's manager—with an approval process. When selected, the field is available as an assigned approver option for approval steps. You can always select a hierarchy field here but not use it for any approval steps.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Set **Next Automated Approver Determined By** with one of the following options.

Option	Description
--None--	Approval requests aren't automatically assigned based on a field. Instead, you manually specify a user to approve all approval requests.
`Field`	Approval requests are assigned to an approver from the specified field. You can select only a hierarchical relationship field, such as `Manager`.
Use Approver Field of *`Object`* **Owner**	Available only when *`Field`* is selected. If selected, the first executed approval step sets the approver to the value of *`Field`* on the record owner's user record—instead of the submitter's user record.

Option	Description
	All remaining steps use *Field* in the user record of the preceding step's approver.

Example: If you select the Manager field, you can configure any step in this process to route approval requests to the submitting user's manager.

If you select **Use Approver Field of** *Object* **Owner**, the first step that isn't skipped is routed to the owner's manager. All other steps are routed to the previous approver's manager.

SEE ALSO:

Custom Field Types

Considerations for Setting Approvers

Specify Who Can Edit Locked Records

When a record is submitted for approval, it's locked to prevent users from editing it during the approval process. Use the record editability properties to determine who can edit records that are locked in this approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Option	Description
Administrators ONLY...	Default. Lets users edit the record that's pending approval only if they have: • The "Modify All" object-level permission for the given object, or • The "Modify All Data" permission
Administrators OR...	Lets the assigned approver and admins edit the record.

Note: Even when a campaign is locked for approval, users can add campaign members to it.

Design the Approval Request Page

The approval page is where an approver responds to an approval request. Customize which fields appear on that page and in which order. This page is used only for this approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Option	Description
Selected Fields	Specifies which fields to display on the approval request page. Keep in mind that approvers could view this page on a mobile device. Select only the fields necessary for users to decide whether to approve or reject records.
Display approval history information...	If selected, the approval request page displays the approval history of the associated record.

Option	Description
Security Settings	Controls whether users have to log in to Salesforce to see the approval request. **Allow approvers to access the approval page only from within the application (Recommended)** Default. Users log in to Salesforce to view the approval page. **Allow approvers to access the approval page only from within the application, or externally from a wireless-enabled mobile device** Users can access an external version of the approval page from a browser, including browsers on mobile devices, without logging in to Salesforce. If selected, you can't add approval steps that let users manually select the next approver.

Specify Who Can Submit Records to an Approval Process

Only specified individuals or roles can submit a record for approval. You can also let submitters recall an approval request.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Submitter Type	Select a type or search to populate the Available Submitters list.
Allowed Submitters	If the user who submits a record for approval isn't included in this list, the record doesn't enter this approval process—even if the record meets the entry criteria.
Allow submitters to recall approval requests	If selected, submitters can recall their approval requests. If unselected, only admins can recall requests. This option is useful for situations where things can change on the submitter's side while waiting for an approval. For example, an opportunity could be lost after the user submits it for approval.

Add an Approval Step to an Approval Process

Approval steps define the chain of approval for a particular approval process. Each step determines which records can advance to that step, who to assign approval requests to, and whether to let each approver's delegate respond to the requests. The first step specifies what to do if a record doesn't advance to that step. Later steps specify what happens if an approver rejects the request.

You can add steps to an approval process only if it is inactive.

From the approval process, click **New Approval Step**, and follow the wizard.

Steps are executed in the order specified.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval steps:
- "Customize Application"

IN THIS SECTION:

1. Control Which Records Apply to an Approval Step
   Control which records are part of the approval step by setting the step's criteria. You can also specify what happens to records that don't meet the step's criteria.
2. Identify Assigned Approvers for an Approval Step
   Specify who to send an approval request for this step to.

3. Specify Rejection Behavior for an Approval Step
   Configure what happens if an approver rejects a request. The final rejection actions for the approval process determine the first step's rejection behavior.

SEE ALSO:

- Set Up an Approval Process
- Review an Approval Process
- Enable Email Approval Response

Control Which Records Apply to an Approval Step

Control which records are part of the approval step by setting the step's criteria. You can also specify what happens to records that don't meet the step's criteria.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Criteria Options

If all records go through this approval step, leave **All records should enter this step** selected.

If only certain types of records should enter this process, select **Enter this step if the following...** and choose the appropriate option **(1)**. For details on the options, see Control Which Records Apply to an Approval Process.

(2) Else Options for Approval Step Criteria

If you specified filter criteria or entered a formula, choose what happens to records that do not meet the criteria or if the formula does not return `True`.

Note: You can't change your selection after the approval process has been activated, even if you deactivate the approval process.

Option	Description
approve record	Approves the request and performs all final approval actions.
reject record	Rejects the request and performs all final rejection actions. This option is available only for the first step in the approval process.
go to next step	Skips this step and goes to the next step. Available only when there's a later step. When you apply this option in the first step, keep in mind: • If the record doesn't meet the criteria for any subsequent steps, the record is rejected. • If you delete all later steps, Salesforce rejects the record. When you apply this option in another step, keep in mind:

Option	Description
	• If you delete all later steps, Salesforce ends the process.

SEE ALSO:

Set Up an Approval Process
Review an Approval Process
Enable Email Approval Response

Identify Assigned Approvers for an Approval Step

Specify who to send an approval request for this step to.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Select Approver	Specify who to assign the approval to. **Let the submitter choose the approver manually. (default)** Prompts the user to select the next approver. **Automatically assign an approver using a standard or custom hierarchy field.** Assigns the approval request to the user in the field displayed next to this option. You select this field when you configure the approval process. **Automatically assign to a queue.** Available only for objects that support queues. Assigns approval requests to a queue. **Automatically assign to approver(s).** Assigns the approval request to one or more specific users, specific queues, or users related to the submitted record. You can add up to 25 per step.
When multiple approvers are selected:	Available only when `Automatically assign to approver(s)` is selected. **Approve or reject based on the first response.** The first response to the approval request determines whether the record is approved or rejected. **Require unanimous approval from all selected approvers.** The record is approved only if everyone approves the request. If any approvers reject the request, the approval request is rejected.
The approver's delegate may also approve this request	Users can identify a delegate in their approval settings. Delegated approvers can't reassign approval requests; they can only approve or reject them.

SEE ALSO:

Considerations for Setting Approvers

Specify Rejection Behavior for an Approval Step

Configure what happens if an approver rejects a request. The final rejection actions for the approval process determine the first step's rejection behavior.

Option	Description
Perform all rejection actions...	Rejects the request completely even if previous steps were approved. Salesforce performs all rejection actions specified for this step and all final rejection actions.
Perform ONLY the rejection actions for this step...	Rejects the request, and returns the approval request to the previous approver. Salesforce performs all rejection actions specified for this step.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Add Automated Actions to an Approval Process

You can associate actions to approval steps, initial submission, final approval, final rejection, or recall. Approval processes support four automated actions.

Action Type	Description
Task	Assigns a task to a user who you specify. You can specify the subject, status, priority, and due date of the task.
Email Alert	Sends an email to a designated recipient using a specified email template.
Field Update	Changes the value of a selected field. You can specify a value or create a formula for the new value.
Outbound Message	Sends a message to a designated endpoint. You can also specify a username and the data to include in the message. Not supported for approval processes on junction objects.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: When expenses are approved, you want to print checks for payment. To do so, you add an outbound message, which sends the appropriate information to your Oracle accounting service, as a Final Approval action.

IN THIS SECTION:

Groups of Automated Actions in an Approval Process

Each approval process is organized into groups of actions based on when the actions occur, such as initial submission. To add an automated action to your approval process, determine which group of actions to add it to.

Add an Automated Action to Your Approval Process

If you didn't create an automated action before configuring your approval process, you can create one directly from the approval process.

Add an Existing Automated Action to Your Approval Process

If you've already created an automated action, you can add it to your approval process.

SEE ALSO:

Set Up an Approval Process

Automated Actions

Considerations for Automated Actions

Groups of Automated Actions in an Approval Process

Each approval process is organized into groups of actions based on when the actions occur, such as initial submission. To add an automated action to your approval process, determine which group of actions to add it to.

Group	Occurs When...	Default Actions
Initial Submission	A user first submits a record for approval.	Record Lock (locks)
Approval Step Approval	All required approvals for this step have been given for a record.	None
Approval Step Rejection	An approver rejects this request for this step.	None
Final Approval	All required approvals have been given for a record	Record Lock (locks)
Final Rejection	An approver rejects the request, and it goes to the final rejection state.	Record Lock (unlocks)
Recall	A submitted approval request is recalled.	Record Lock (unlocks

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Considerations for Automated Actions

Add an Automated Action to Your Approval Process

If you didn't create an automated action before configuring your approval process, you can create one directly from the approval process.

1. Open the approval process that you want to add an action to.
2. From the appropriate related list, click **Add New**. For an approval step where the Approval Actions and Rejection Actions are hidden, click **Show Actions**.
3. Choose the type of action.

   The list of available actions differs depending on your settings and whether you have reached the limit for a type of action.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval actions:

- "Customize Application"

4. Configure the action.

SEE ALSO:

- Set Up an Approval Process
- Considerations for Automated Actions
- Groups of Automated Actions in an Approval Process

Add an Existing Automated Action to Your Approval Process

If you've already created an automated action, you can add it to your approval process.

1. Open the approval process that you want to add an action to.
2. From the appropriate related list, click **Add Existing**. If that button is hidden, click **Show Actions**.
3. Choose the type of action.
4. Move the action from Available Actions to Selected Actions.
5. Save your changes.

SEE ALSO:

- Groups of Automated Actions in an Approval Process
- Considerations for Automated Actions

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To select approval actions:

- "Customize Application"

Review an Approval Process

Generate a graphical representation of your approval process before you activate it. Use the Process Visualizer to gain buy-in from step owners and reinforce your company's policies by documenting the decisions you reached when the approval process was designed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Process Visualizer Requirements
Before you can use the Process Visualizer, confirm that your browser and org meet the requirements.

Tour the Process Visualizer User Interface
The user interface for the Process Visualizer has several functional areas.

Process Visualizer Notation
The Process Visualizer uses specific notation to graphically represent each approval process. The shapes are based on the Business Process Modeling Notation (BPMN) standard. In the Process Visualizer, a shape's color changes from gray to blue when you hover over or click it.

Open an Approval Process in the Process Visualizer
To see a graphical representation of an approval process, open it in the Process Visualizer.

Print Your Approval Process Diagram
Generate a printed copy or PDF file of your approval process to communicate information about the approval process. Sharing the diagram promotes buy-in from the people who participate in the approval decisions and helps approval process users understand the requirements and flow.

Navigate a Large Approval Process Diagram

If you're reviewing a large approval process in the Process Visualizer, we've provided tools to make it easier for you.

SEE ALSO:

Set Up an Approval Process

Activate an Approval Process

Process Visualizer Requirements

Before you can use the Process Visualizer, confirm that your browser and org meet the requirements.

- Your browser must have the Adobe Flash Player plug-in, version 9.0.115 or later.
- Your org must have at least one approval process defined.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Open an Approval Process in the Process Visualizer

Review an Approval Process

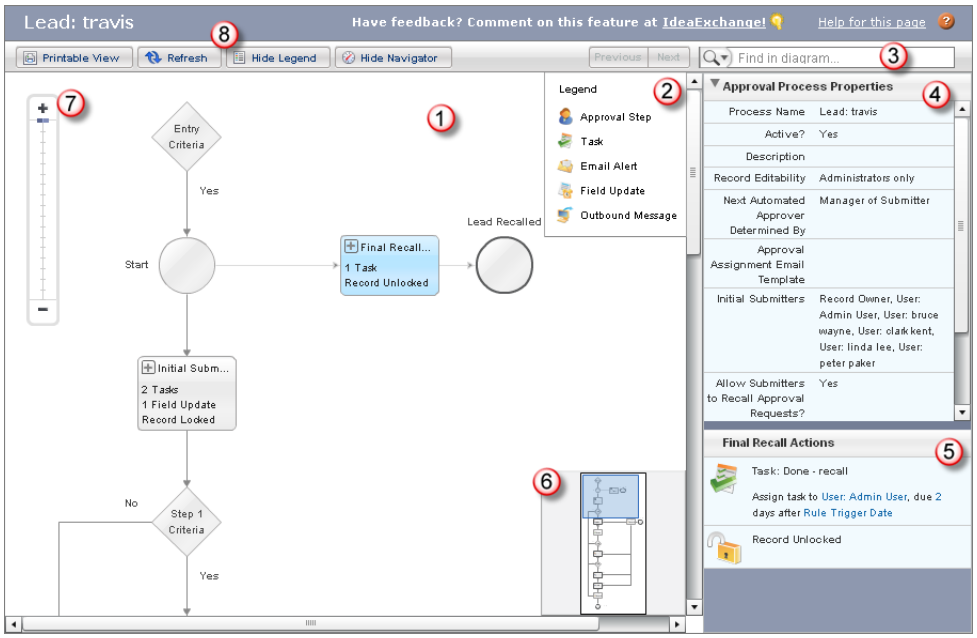
Tour the Process Visualizer User Interface

The user interface for the Process Visualizer has several functional areas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



The highlighted numbers in the illustration correspond to the following descriptions.

(1) Canvas

View the approval process diagram. Entry criteria appear at the top of the diagram. Record processing moves down the flowchart until the final approval or rejection. Each flowchart component uses a shape, an icon, or both to represent the type of processing that could happen.

- **(6) Navigator window**: Drag the pane within the navigator to view different parts of the diagram.
- **(7) Zoom**: Zoom in and out of your process diagram.

Sidebar

- **(4) Approval Process Properties**: View details about the approval process.
- **(5) Hover details**: View details about the component on the diagram that's selected.

Button bar

- **(8)** Print the diagram, refresh the diagram, or toggle the legend or navigator.
- **(3)** Find specific text in the diagram.

SEE ALSO:

Open an Approval Process in the Process Visualizer

Process Visualizer Notation

Review an Approval Process

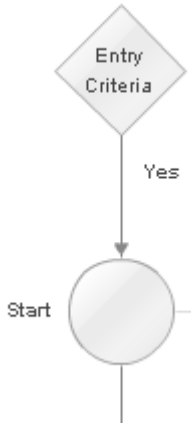
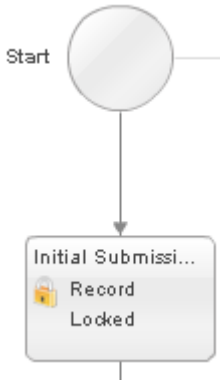
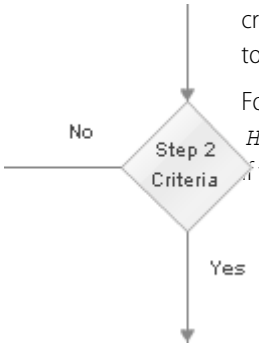
Process Visualizer Notation

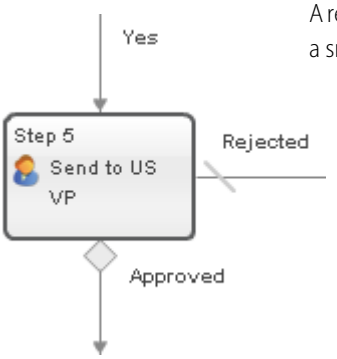
The Process Visualizer uses specific notation to graphically represent each approval process. The shapes are based on the Business Process Modeling Notation (BPMN) standard. In the Process Visualizer, a shape's color changes from gray to blue when you hover over or click it.

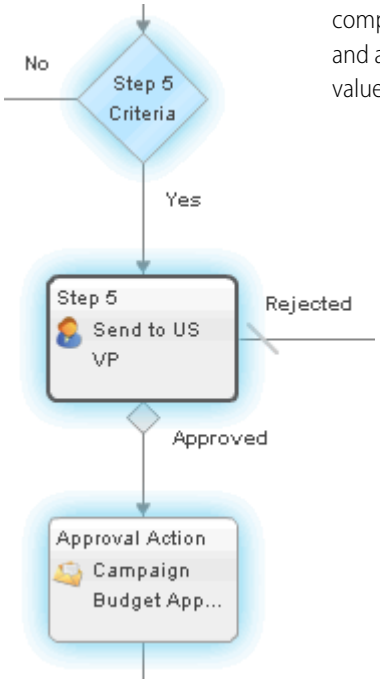
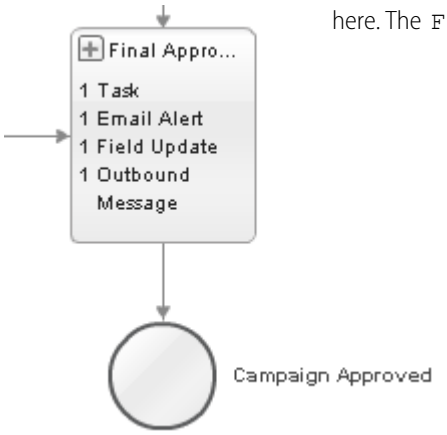
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Icon and Shape Examples	Description
	Near the top of the diagram, a `Start` circle indicates the beginning of the approval process. If defined, an `Entry Criteria` diamond precedes the `Start` circle. When you hover over or click `Entry Criteria`, the Process Visualizer displays the entry criteria in the sidebar. For example, an approval process for a lead is triggered when *`Lead.AnnualRevenue >= 10000`*. The arrow below the `Start` circle leads to the next criteria in the approval process. If a recall action was defined for the entry criteria, a line to the right connects to the `Final Recall Actions` rectangle.
	The rectangle labeled `Initial Submission Actions` represents any actions you defined in initial submission. At a minimum, this element shows the record lock set automatically by Salesforce to prevent updates by other users.
	A diamond with `Yes` or `No` branches represents the numbered step's criteria or formula that is evaluated before proceeding. Click the diamond to see more information in the sidebar. For example, the criteria determines whether *`Lead.Rating = Hot`*. If true, the record continues to the next step on the `Yes` branch. If false, it continues to the next step on the `No` branch.

Icon and Shape Examples	Description
	A rectangle containing a human icon ( ) represents an approval step. The Approved path includes a small diamond. The Rejected path uses a backslash.
	A rectangle without a human icon or a plus sign represents a single action. The action types and icons are: • Send an email alert: • Start a task: • Perform a field update: • Send an outbound message: In this example, a task has been assigned. When you hover over or click the rectangle, look for more information in the sidebar.
	A rectangle with a plus sign (+) represents multiple actions. The actions shown inside are performed simultaneously as a unit. To view the individual actions, click the plus sign.

Icon and Shape Examples	Description
	When you hover over or click a shape on the diagram, and it is part of a step, all the elements that comprise the step use a glowing background color. This visual clue helps you locate all the decisions and actions that comprise each step, and how the processing may branch based on the record's values.
	In the approval process definition, if the option was enabled to perform only the rejection actions for this step and send the approval request back to the most recent approver (go back one step), the diagram displays a `Go Back` circle. When you click the `Go Back` circle, a message displays in the sidebar. For example: `On rejection send the request back to any one of the following steps, depending on the most recent approver: - list-of-steps. If no approvers are found, the request goes to Final Rejection.` The identified steps depend on where in the approval process the Go Back was defined. The possible steps are highlighted when you hover over the `Go Back` circle. In the diagram, the Process Visualizer displays the `Go Back` circle only for a step where its preceding step is defined as a skip.
	The `Final Approval Actions` rectangle leads to a circle labeled `Approved`, as shown here. The `Final Rejection Actions` rectangle leads to a circle labeled `Rejected`. If a

Icon and Shape Examples	Description
	recall action was defined, the `Final Recall Actions` rectangle shows that the record is now unlocked, and the arrow leads to a circle labeled `Recalled`.

SEE ALSO:

- Open an Approval Process in the Process Visualizer
- Tour the Process Visualizer User Interface
- Review an Approval Process

Open an Approval Process in the Process Visualizer

To see a graphical representation of an approval process, open it in the Process Visualizer.

1. From Setup, enter *`Approval Processes`* in the `Quick Find` box, then select **Approval Processes**.
2. Click the name of the approval process you want to view.
3. On the detail page for the approval process, click **View Diagram**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view approval steps:

- "View Setup and Configuration"

SEE ALSO:

- Tour the Process Visualizer User Interface
- Process Visualizer Notation
- Review an Approval Process

Print Your Approval Process Diagram

Generate a printed copy or PDF file of your approval process to communicate information about the approval process. Sharing the diagram promotes buy-in from the people who participate in the approval decisions and helps approval process users understand the requirements and flow.

1. Open the approval process in the Process Visualizer.
2. Click **Printable View**.
   In this view, the numbers on the diagram correspond to details shown in a table below the diagram.
3. Click **Print**, and follow your browser's print dialog. If you have an Adobe PDF print driver installed, you can save the printable view as a PDF file.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To print an approval process diagram:

- "View Setup and Configuration"

SEE ALSO:

- Process Visualizer Notation
- Review an Approval Process

Navigate a Large Approval Process Diagram

If you're reviewing a large approval process in the Process Visualizer, we've provided tools to make it easier for you.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Find Specific Text

Find matching text within the diagram and the Approval Process Properties pane, and view the results highlighted in the sidebar. The Find feature is helpful when viewing a diagram with 10 or more steps, complex formulas, or the names of many people or roles on the approvers" lists. Find is case insensitive and treats the search terms as a phrase in quotes.

- Scroll through the matches by clicking the **Previous** and **Next** buttons.
- Limit results by selecting a filter: Approval Steps, Tasks, Email Alerts, Field Updates, and Outbound Messages. If you select a filter but don't enter any text, all instances of that filter item are highlighted in the diagram and navigation pane. **All** is the only filter that includes Approval Process Properties, Entry Criteria, and Go Back steps in its search parameters.

Zoom

Change the diagram's size with the zoom slider. Zoom is visible only for diagrams that are bigger than the display window.

Navigator Window

Quickly view different areas of a large approval process diagram by clicking and dragging the pane around the navigator window. When you use the Find feature, the navigator window changes to highlight the matches and shows items not visible in the main diagram display area. The navigator also reflects changes to the size of the diagram in the main display when you use the zoom slider.

SEE ALSO:

Tour the Process Visualizer User Interface

Review an Approval Process

Activate an Approval Process

After you've created at least one step for the approval process, activate the process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To activate approval processes:

- "Customize Application"

1. Open the approval process.
2. Make sure that it's configured correctly.
3. Click **Activate**.

SEE ALSO:

Review an Approval Process

Prepare Your Org for Approvals

Considerations for Managing Approvals

Prepare Your Org for Approvals

Make sure that your users can submit their records for approval, and consider how you can make it easy for approvers to respond to approval requests.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Let Users Submit for Approval

After you activate an approval process for an object, customize the object's page layouts to support record submission.

Let Users Respond to Approval Requests from the Home Page in Salesforce Classic

Add the Items To Approve related list to your custom home page layouts to give users an instant view of their approval requests.

Let Users Respond to Approval Requests by Email

If the email notification includes all the information that an approver needs to decide, enable email approval response. That way, a user can simply reply to the email notification.

Let Users Respond to Approval Requests from Chatter

If your users don't need in-depth information to decide how to respond to an approval request, enable Approvals in Chatter. That way, they don't have to leave their feed to continue with their day-to-day tasks.

SEE ALSO:

Set Up an Approval Process

Considerations for Approvals

Let Users Submit for Approval

After you activate an approval process for an object, customize the object's page layouts to support record submission.

Add the following components to your page layouts.

- Submit for Approval button
- Approval History related list

  The Approval History related list lets users submit approval requests and track a record's progress through an approval process from the record detail page.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To modify page layouts:

- "Customize Application"

SEE ALSO:

Page Layouts

Prepare Your Org for Approvals

Let Users Respond to Approval Requests from the Home Page in Salesforce Classic

Add the Items To Approve related list to your custom home page layouts to give users an instant view of their approval requests.

Do the following for each user receiving approval requests.

1. From Setup, enter *Home Page Layouts* in the Quick Find box, then select **Home Page Layouts**.
2. Next to the appropriate home page layout, click **Edit**.
3. Select **Items to Approve**.
4. Complete the Home Layout wizard, and save your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To modify home page layouts:
- "Customize Application"

SEE ALSO:

Customizing Home Tab Page Layouts

Prepare Your Org for Approvals

Let Users Respond to Approval Requests by Email

If the email notification includes all the information that an approver needs to decide, enable email approval response. That way, a user can simply reply to the email notification.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Considerations for Email Approval Response

Before you enable the ability to act on approvals via email, review how email works with your approval processes.

Default Template for Email Approval Response

When you enable email approval response, Salesforce uses a default email template for approval processes—unless you specify a custom email template.

Enable Email Approval Response

After you've reviewed the considerations and prepared the right template, flip the switch that lets users respond to approval requests directly from their email.

SEE ALSO:

Prepare Your Org for Approvals

Let Users Respond to Approval Requests from Chatter

Considerations for Email Approval Response

Before you enable the ability to act on approvals via email, review how email works with your approval processes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Compatibility with Approval Processes

Email approval response isn't supported for approval processes that:

- Assign approval to a queue
- After the first step, let the approver manually select the next approver

Implicit Agreement with Salesforce

By enabling the email approval response feature, you agree to let Salesforce:

- Process email approval responses
- Update approval requests for all active users in your org
- Update the approval object on behalf of your org's users

SEE ALSO:

Considerations for Approvals

Let Users Respond to Approval Requests by Email

Default Template for Email Approval Response

When you enable email approval response, Salesforce uses a default email template for approval processes—unless you specify a custom email template.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
Requesting User has requested your approval for the following item.

To approve or reject this item, reply to this email with the word APPROVE, APPROVED, YES,
 REJECT, REJECTED, or NO
in the first line of the email message, or click this link:

Link to approval request page

If replying via email you can also add comments on the second line. The comments will be
stored with the approval request in Salesforce CRM.

Note: For salesforce.com to process your response the word APPROVE, APPROVED, YES, REJECT,
 REJECTED, or NO must be in the very first line of the reply email. Also, any comment must
 be in the second line.
```

If your org has Approvals in Chatter enabled and the approver opted to receive notifications as Chatter posts, the default email template is appended with:

```
You can also approve, reject and comment on this request from your Chatter feed:

Link to approval post in Chatter
```

Note: If you use a custom email template for your approval process, make sure that it explains both response options: clicking the link and replying by email. If the user doesn't respond correctly (for example, if the user misspells approve or types it on the wrong line), Salesforce doesn't register the response.

SEE ALSO:

Merge Fields for Approvals

Let Users Respond to Approval Requests by Email

Enable Email Approval Response

After you've reviewed the considerations and prepared the right template, flip the switch that lets users respond to approval requests directly from their email.

Before you begin, give the appropriate users the "API Enabled" user permission so that they can respond to approval requests by email.

1. From Setup, enter *Process Automation Settings* in the Quick Find box, then select **Process Automation Settings**.
2. Select **Enable Email Approval Response**.
3. Save your changes.

SEE ALSO:

Considerations for Email Approval Response

Let Users Respond to Approval Requests by Email

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To enable Email Approval Response:

- "Customize Application"

Let Users Respond to Approval Requests from Chatter

If your users don't need in-depth information to decide how to respond to an approval request, enable Approvals in Chatter. That way, they don't have to leave their feed to continue with their day-to-day tasks.

IN THIS SECTION:

Prepare to Enable Approvals in Chatter

Because Approvals in Chatter relies on both Chatter and the Approvals feature, getting your org set up involves more than just turning on the feature. Before you enable Approvals in Chatter, understand the limitations and considerations for Approvals in Chatter and post templates.

Considerations for Approvals in Chatter

Before you enable Approvals in Chatter, understand how it works.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Enable Approvals in Chatter

If your organization has both Approvals and Chatter enabled, administrators can turn on Approvals in Chatter, which lets users receive approval requests as posts in their Chatter feeds.

Where Do Approval Request Posts Appear?

When your org has Approvals in Chatter enabled, approval request posts appear in various Chatter feeds. To see the approval request post, you must have access to the approval record.

Chatter Post Templates for Approval Requests

Approval post templates for Chatter let you customize the information that is included in the approval request post when it displays in a Chatter feed.

Prepare to Enable Approvals in Chatter

Because Approvals in Chatter relies on both Chatter and the Approvals feature, getting your org set up involves more than just turning on the feature. Before you enable Approvals in Chatter, understand the limitations and considerations for Approvals in Chatter and post templates.

Do the following for each object for which you want approval requests to appear in Chatter.

1. Enable feed tracking.
2. Create an approval post template.

   Tip: For each object, create one post template that can apply for all approval processes. Mark that post template the default for the object.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Chatter Post Templates for Approval Requests

Where Do Approval Request Posts Appear?

Considerations for Approvals in Chatter

Considerations for Approvals in Chatter

Before you enable Approvals in Chatter, understand how it works.

- When Approvals in Chatter is enabled in your org, it is turned on for all users. Users can then update their own Chatter settings to opt out of receiving approval requests as posts in their Chatter feeds.
- Chatter post approval notifications are available only for approval processes associated with an object that has been enabled for feed tracking.
- If the approval object is a detail object in a master-detail relationship, `Owner` isn't available for approval page layouts or approval post templates.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Limitations

- Approvals in Chatter doesn't support delegated approvers or queues.
- You can't recall or reassign an approval request from a post. Instead, perform these actions from the approval record.
- Approval requests from Sites or portal users aren't supported.

Approval Posts

- Approval posts can't be deleted in the Salesforce user interface; you can only delete them through the API.

- If you don't select an approval post template, the approval post for the object uses either the system default template or the default template for the object if you created one.
- Only users with access to the approval record can see the approval request post. Comments on approval posts aren't persisted to the approval record.
- Different users see different configurations of the approval request post.
  - Only approvers see approval action buttons on their posts, and then only in their profile feed or their news feed.
  - Only approvers see approver names in the header.
- If you change the approver, step name, or the routing type on an approval process while it's in progress, existing approval posts aren't updated.
- When an approval request is recalled, a new post is generated. It appears on the news feeds of the submitter, all approvers, and followers of the object. It also appears on the record feed.
- If a step requires unanimous approval from multiple approvers, the approval request post for that step doesn't list all selected approvers in its header. Approvers see only their own name in the post header.

SEE ALSO:

Let Users Respond to Approval Requests by Email

Prepare to Enable Approvals in Chatter

Where Do Approval Request Posts Appear?

Considerations for Approvals

Enable Approvals in Chatter

If your organization has both Approvals and Chatter enabled, administrators can turn on Approvals in Chatter, which lets users receive approval requests as posts in their Chatter feeds.

Before you begin, make sure that all approval processes in your org are properly configured to take advantage of Approvals in Chatter. After turning this on feature, all existing active approval processes start generating Chatter posts.

1. From Setup, enter *Chatter Settings* in the Quick Find box, then select **Chatter Settings**.
2. Click **Edit**.
3. Select **Allow Approvals**.
4. Save your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To enable or Approvals in Chatter:

- "Customize Application"

SEE ALSO:

Prepare to Enable Approvals in Chatter

Considerations for Approvals in Chatter

Where Do Approval Request Posts Appear?

Where Do Approval Request Posts Appear?

When your org has Approvals in Chatter enabled, approval request posts appear in various Chatter feeds. To see the approval request post, you must have access to the approval record.

Approval request posts show up in these feeds.

- Chatter feed of the assigned approver
- Submitter's profile
- Chatter feed of the submitter if the submitter is following the approval request record
- Chatter feed of the approval request record
- Chatter feed of anyone following the approval request record
- Object-specific filter on the Chatter feed of anyone following the approval record
- Company filter of every user with access to the approval record

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

What Happens When You Opt Out of Chatter Approval Requests?
Considerations for Approvals in Chatter
Let Users Respond to Approval Requests from Chatter

Chatter Post Templates for Approval Requests

Approval post templates for Chatter let you customize the information that is included in the approval request post when it displays in a Chatter feed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Considerations for Chatter Post Templates for Approval Requests
Keep these limitations and dependencies in mind when working with post templates.

Create a Chatter Post Template
Identify which fields to display in an approval request post.

SEE ALSO:

Create a Chatter Post Template
Manage Deleted Custom Fields

Considerations for Chatter Post Templates for Approval Requests

Keep these limitations and dependencies in mind when working with post templates.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Limitations

- The associated object must be enabled for approvals and feed tracking.
- You can't delete an approval post template if it's in use by an approval process.

Dependencies

- Deleting a custom field removes it from any approval post template that references it. Existing posts aren't affected. Undeleting the custom field restores it to the available fields list, but doesn't restore it to any approval post templates that previously contained it.

- Deleting (or undeleting) a custom object also deletes (or undeletes) its associated approval post templates and any of its approval request posts that are already in Chatter feeds.
- If you rename a custom object, approval post templates associated with it update accordingly.

SEE ALSO:

Create a Chatter Post Template

Considerations for Approvals

Create a Chatter Post Template

Identify which fields to display in an approval request post.

1. From Setup, enter *Post Templates* in the Quick Find box, then select **Post Templates**.
2. Click **New Template**.
3. Select the object for your template.
4. Click **Next**.
5. Give the template a name and description.
6. If you want this template to be the default for the associated object, select **Default**.
7. Add up to four fields to display on the approval request post.

   We recommend putting any text-heavy fields—such as Comments or Description—at the bottom.
8. Save your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create approval request post templates:

- "Customize Application"

SEE ALSO:

Choose Approval Request Notification Templates

Considerations for Chatter Post Templates for Approval Requests

Considerations for Approvals

Before you automate something with an approval process, be aware of the limitations of the feature.

Users can't see which approval process is triggered when they click **Submit for Approval**. Familiarize users on the criteria for each approval process and what each approval process does. If the record doesn't meet the entry criteria or if they're not an allowed submitter for any approval processes, Salesforce displays an error.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Considerations for Configuring Approvals

When creating or editing an approval process, keep in mind how approvals are compatible with other features. Before you start, draw out the steps of your approval process.

Merge Fields for Approvals

Approval merge fields include *{!ApprovalRequest.fieldName}* and *{!ApprovalRequestingUser.fieldName}*. They're supported in certain email templates and return different values based on the status of the approval process instance.

Considerations for Setting Approvers

When you specify approvers for a given approval step—or for the only step if you're using the jump start wizard—keep some things in mind.

Considerations for Managing Approvals

Keep these things in mind when maintaining existing approval processes—including activating and deleting them.

SEE ALSO:

Considerations for Email Approval Response

Considerations for Approvals in Chatter

Considerations for Approval History Reports

Considerations for Configuring Approvals

When creating or editing an approval process, keep in mind how approvals are compatible with other features. Before you start, draw out the steps of your approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Associated Object

If the approval object is a detail object in a master-detail relationship, `Owner` isn't available for approval page layouts or approval post templates.

Approval Criteria

In approval criteria—either the entry criteria or step criteria—don't reference expressions that resolve to random values. That way, if the criteria needs to be evaluated again, the record is evaluated the same every time.

Compatibility with Other Features

- Flows can delete records that are pending approval.
- Design automated actions so that you can use them for both workflow rules and approval processes.

Field Update Actions in Approvals

- An approval process can specify a field update action that reevaluates workflow rules for the updated object. If, however, the re-evaluated workflow rules include a cross-object field update, those cross-object field updates are ignored.
- Field updates that are executed as approval actions don't trigger workflow rules or entitlement processes.

Anticipate Errors

Consider reviewing the content on approvals errors. That way, you can anticipate common issues and configure your approval process so that the error is less likely.

SEE ALSO:

What Does This Approvals Error Mean?

Set Up an Approval Process

Considerations for Automated Actions

Considerations for Chatter Post Templates for Approval Requests

Merge Fields for Approvals

Approval merge fields include `{!ApprovalRequest.fieldName}` and `{!ApprovalRequestingUser.fieldName}`. They're supported in certain email templates and return different values based on the status of the approval process instance.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tip: The submitter isn't always the current user. For custom email templates, use `{!ApprovalRequestingUser.fieldName}` instead of `{!User.fieldName}`.

Where Are Approval Merge Fields Supported?

You can use approval process merge fields in email templates, but not mail merge templates. Except for `{!ApprovalRequest.Comments}`, approval merge fields named `{!ApprovalRequest.field_name}` in email templates return values only in approval assignment emails and email alerts for approval processes. When used in other emails—including email alerts for workflow rules—the approval merge field returns `null`.

What Values Does a Merge Field Provide?

The generated value of an ApprovalRequest merge field depends on which step the approval process is in.

- In the approval request email, a merge field returns the submitter's name and the name of the first step.
- When the request is approved, the merge field returns the most recent approver's name and the name of the second step, if applicable.
- For subsequent actions, a merge field value returns the previous completed step.
- For an approval step that requires unanimous approval from multiple approvers, `{!ApprovalRequest.Comments}` returns only the most recently entered comment in emails.

SEE ALSO:

Default Template for Email Approval Response

Considerations for Setting Approvers

When you specify approvers for a given approval step—or for the only step if you're using the jump start wizard—keep some things in mind.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Users with the following permissions can respond to approval requests, even if they aren't designated approvers.
  - "Modify All Data"
  - "Modify All" for an object
- Make sure that the assigned approver has access to read the records for the approval requests. For example, a user who can't view expense records can't view expense approval requests.
- Approval processes that let users select an approver manually also let users select themselves as the approver.
- You can assign an approval request to the same user multiple times in a single step. However, Salesforce sends the user only one request.
- Here's what happens to the list of approvers after a record enters an approval step and the approval process later returns to that step.
  - If the user who responded **isn't in the designated approvers list and has either "Modify All Data" or "Modify All" permissions for the object**, that user replaces the original approver in the list of approvers.

- If a user who responded **is in the designated approvers list**, the list of approvers for that step don't change. This behavior occurs even if the field values that designate the approvers have changed.

For example, an approval process's first step requests approval from a user's manager. If the approval request is rejected in the second step, the approval request returns to the first step. This table explores what happens to the list of approvers.

If...	The Designated Approver Is...
The user's manager originally responded to the approval request.	The manager
The user's manager originally responded to the approval request. Since then, the user's manager has changed.	The original manager The new manager is not a designated approver for this step.
A user with the "Modify All Data" permission originally responded to the approval request.	The user with the "Modify All Data" permission That user replaces the user's manager in the list of designated approvers for this step.

Assigning Approval Steps to Queues

You can assign approval requests to a queue only if the associated object supports queues. Email approval response isn't supported for approval processes that assign approval to a queue.

When the assigned approver is a queue:

- Any queue member can approve or reject an approval request that is assigned to the queue.
- Approval request emails are sent to the queue email address. If the queue is set up to send email to members, approval request emails get sent to the queue members, unless their approval user preferences are set to never receive approval request emails.
- Because email notifications to a queue aren't intended for an external audience, `{!ApprovalRequest.External_URL}` returns the equivalent internal URL.
- Salesforce1 notifications for approval requests aren't sent to queues. For each approval step involving a queue, we recommend adding individual users as assigned approvers, so at least those individuals can receive the approval request notifications in Salesforce1. To have both queues and individual users as assigned approvers, select `Automatically assign to approver(s)` instead of `Automatically assign to queue` in the approval step.
- When an approval request is rejected and returned to the previous approver and the previous approver was a queue, the approval request is assigned to the user who approved it instead of the queue.
- The Approval History related list displays the queue name in the `Assigned To` column and the actual user who approved or rejected the approval request in the `Actual Approver` column.

SEE ALSO:

Identify Assigned Approvers for an Approval Step

Considerations for Approvals

Considerations for Managing Approvals

Keep these things in mind when maintaining existing approval processes—including activating and deleting them.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Admin Permissions

Users with one of these permissions are considered approval admins.

- "Modify All" object-level permission for the given object
- "Modify All Data" user permission

Approval admins can:

- Approve or reject pending approval requests without being part of the approval process
- Edit records that have been locked for approval

Activating Approval Processes

- An approval process must have at least one step before you can activate it.
- Before you activate your approval process, test it in your Salesforce sandbox.
- After an approval process is activated, you can't add, delete, or change the order of the steps or change its reject or skip behavior, even if the process is inactive.

Monitoring In-Flight Approval Processes

Standard reports for approval requests are included in both the Administrative Reports folder and the Activity Reports folder.

Deleting Approval Processes

Before you delete an approval process:

1. Make sure it's inactive.
2. Delete pending approval requests that are associated with it, and remove them from the Recycle Bin.

SEE ALSO:

Review an Approval Process

Activate an Approval Process

Manage Multiple Approval Requests

Considerations for Approvals

Sample Approval Processes

Review samples of common approval processes to help you get started creating your own.

IN THIS SECTION:

Sample Approval Process: PTO Requests

Most companies require employees to file a PTO (Paid Time Off) request and have their manager approve it. In three phases, here's how to automate a simple one-step PTO request process using Salesforce.

Sample Approval Process: Expense Reports

If your company requires that employees file expense reports for managers to approve, you can automate this process in Salesforce. Use this example to create a two-step expense report approval process for all employees in your headquarters office. It specifies that expenses less than $50 are automatically approved, expenses $50 and over require manager approval, and expenses over $5,000 require additional approval from two VPs. This example highlights a parallel approval process and the "else" option.

Sample Approval Process: Discounting Opportunities

Opportunities that are discounted more than 40% require a CEO approval. Use this example to create a one-step approval process.

Sample Approval Process: Job Candidates

When your company interviews candidates for a position, you may have several levels of approval before you can send an offer letter. Use this example to create a three-step approval process that requires approval from multiple management levels.

Sample Approval Process: PTO Requests

Most companies require employees to file a PTO (Paid Time Off) request and have their manager approve it. In three phases, here's how to automate a simple one-step PTO request process using Salesforce.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Prep Your Organization

Before creating the approval process:

- If you do not yet have a custom object to track your PTO requests, create a custom object and tab called PTO Requests. Add the appropriate fields for your PTO Requests such as `Start Date`, `End Date`, and `Employee Name`.
- Create an email template to notify approvers that an approval request needs to be reviewed. Direct users to the approval page in Salesforce by including approval process merge fields.

Create the Approval Process

Use the jump start wizard to create an approval process for the PTO Request custom object and specify the following:

- Select the email template you created for this approval process.
- Don't specify filter criteria. That way, PTO requests are included in this approval process regardless of their attributes.
- Select the `Automatically assign an approver using a standard or custom hierarchy field` option, then choose `Manager`.
- The jump start wizard automatically chooses the record owner as the only person who can submit PTO requests.

Tip: To let the submitter withdraw a submitted PTO request:

1. Click **Edit** and choose **Initial Submitters**.
2. Select `Allow submitters to recall approval requests`.

Wrap Things Up

After creating the approval process, add the Approval History related list to the PTO Request object page layout.

Tip: Consider adding the Items To Approve related list to your custom home page layouts. It gives users an instant view of the approval requests that are waiting for their response.

If available, use your sandbox to test the approval process, then activate it.

SEE ALSO:

Define a Custom Object

Create an Approval Process with the Jump Start Wizard

Prepare Your Org for Approvals

Sample Approval Process: Expense Reports

If your company requires that employees file expense reports for managers to approve, you can automate this process in Salesforce. Use this example to create a two-step expense report approval process for all employees in your headquarters office. It specifies that expenses less than $50 are automatically approved, expenses $50 and over require manager approval, and expenses over $5,000 require additional approval from two VPs. This example highlights a parallel approval process and the "else" option.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Prep Your Organization

Before creating the approval process:

- If you do not yet have a custom object to track your expenses, create a custom object and tab called Expense Reports. Add the appropriate fields such as `Amount`, `Description`, `Status`, `Start Date`, and `End Date`.
- Create an email template to notify approvers that an approval request needs to be reviewed. Direct users to the approval page in Salesforce by including approval process merge fields.
- Create a custom field on the user object called `Office Location`. Assign the "HQ" value to users in the headquarters office location.

Create the Approval Process

Create an approval process using the Expense Report custom object and specify the following:

- The filter criteria for this approval process is *`Current User: Office Location equals HQ`*. Records must meet this criteria before they can be submitted to this approval process.
- Choose the `Manager` field as the next automated approver.
- Create an email template to notify approvers that an approval request needs to be reviewed. Direct users to the approval page in Salesforce by including approval process merge fields.
- Choose the record owner or any other user who you want to be able to submit expense reports.
- Create two approval steps:

  1. Create a step named *`Step 1: Manager Approval`* with the following specifications:
     - Name this step *`Step 1: Manager Approval`*.
     - Select `Enter this step if the following` and choose **criteria are met**. Also, choose **approve record** for the `else` option.
     - Set the filter criteria to: *`Expense: Amount greater or equal 50`*.
     - In the `Automatically assign to approver(s)` option, select the manager of the user submitting the request.
     - If appropriate, choose `The approver's delegate may also approve this request` if you want to allow the user in the `Delegated Approver` field to approve requests.
  2. Create a second approval step named *`Step 2: Multiple VP Approval`* and specify the following:
     - Use the filter criteria *`Expense Amount greater or equal 5000`*.
     - Choose `Automatically assign to approver(s)` and select two users with a VP role.
     - Select the `Require UNANIMOUS approval from all selected approvers` option. The request isn't approved unless both designated users approve.

  - If appropriate, choose `The approver's delegate may also approve this request` if you want to allow the user in the `Delegated Approver` field to approve requests.
  - Choose `Perform ONLY the rejection actions for this step...` so that the request returns to the manager for changes if one of the VPs rejects the request.

Tip: Consider creating the following final approval actions:

- Define a field update to automatically change the `Status` field to "Approved."
- Send an approval notification to the user who submitted the expense report.
- Send an outbound message to your back-office financial system to print a reimbursement check.

Wrap Things Up

After creating the approval process, add the Approval History related list to the Expense Report object page layout.

Tip: Consider adding the Items To Approve related list to your custom home page layouts. It gives users an instant view of the approval requests that are waiting for their response.

If available, use your sandbox to test the approval process, then activate it.

SEE ALSO:

Define a Custom Object
Create Custom Fields
Set Up an Approval Process
Prepare Your Org for Approvals

Sample Approval Process: Discounting Opportunities

Opportunities that are discounted more than 40% require a CEO approval. Use this example to create a one-step approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Prep Your Organization

Before creating the approval process:

- Create an email template to notify approvers that an approval request needs to be reviewed. Direct users to the approval page in Salesforce by including approval process merge fields.
- Create the following custom fields for opportunities:
  - A percent field called `Discount Percent` so that users can enter a percentage discount.
  - A checkbox field called `Discount Approved` to indicate whether the CEO approved the discount.

Create the Approval Process

Create an approval process on the Opportunity object and specify the following:

- The filter criteria for this approval process is *`Discount Percent greater or equal 0.4`*. Records must meet this criteria before they can be submitted to this approval process.

- You don't need to choose a custom field as the next automated approver because you specify later that the CEO must approve all requests.
- Select the email template you created for this approval process.
- Choose the record owner as the only user who can submit a discount request for approval.
- Create one approval step with no filter criteria since all records submitted need to be approved or rejected.
- Choose `Automatically assign to approver(s)` and select the name of your CEO.
- If appropriate, choose `The approver's delegate may also approve this request` if you want to allow the user in the `Delegated Approver` field to approve requests.
- Consider creating the following final approval actions:
  - Email alert to notify the user who submitted the discount request.
  - Field update to automatically select the opportunity `Discount Approved` checkbox.

Wrap Things Up

After creating the approval process, add the Approval History related list to the appropriate opportunity page layouts.

Tip: Consider adding the Items To Approve related list to your custom home page layouts. It gives users an instant view of the approval requests that are waiting for their response.

If available, use your sandbox to test the approval process, then activate it.

SEE ALSO:

Create Custom Fields

Set Up an Approval Process

Prepare Your Org for Approvals

Sample Approval Process: Job Candidates

When your company interviews candidates for a position, you may have several levels of approval before you can send an offer letter. Use this example to create a three-step approval process that requires approval from multiple management levels.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Prep Your Organization

Before creating the approval process:

- If you don't yet have a custom object to track candidates, create a custom object and tab called Candidates. Add the appropriate fields such as `Salary`, `Offer Extended` (checkbox), and `Date of Hire`.
- Create an email template to notify approvers that an approval request needs to be reviewed. Direct users to the approval page in Salesforce by including approval process merge fields.

Create the Approval Process

Create an approval process on the Candidate custom object using the following specifications:

- Don't enter filter criteria because you want all submitted offers to be approved.
- Choose the `Manager` field as the next automated approver.

- Select the email template you created for this approval process.
- Choose the record owner or any other user that you want to be able to submit offer letters.
- Create three approval steps:
  1. Create a step named *Step 1: Manager Approval*:
     - No filter is necessary as you want all records to advance to this step.
     - In the Automatically assign to approver(s) option, select the manager of the user submitting the request.
     - If appropriate, choose The approver's delegate may also approve this request if you want to allow the user in the Delegated Approver field to approve requests.
  2. Create a second step named *Step 2: VP Approval*:
     - No filter is necessary as you want all records to advance to this step.
     - Choose Let the user choose the approver to allow the manager to select the appropriate VP to approve the request.
     - If appropriate, choose The approver's delegate may also approve this request if you want to allow the user in the Delegated Approver field to approve requests.
     - Choose Perform ONLY the rejection actions for this step... so that the request returns to the manager for changes if the VP rejects the request.
  3. Create a third step named *Step 3: CFO Approval*:
     - No filter is necessary as you want all records to advance to this step.
     - Choose Automatically assign to approver(s) and select the name of your CFO.
     - If appropriate, choose The approver's delegate may also approve this request if you want to allow the user in the Delegated Approver field to approve requests.
     - Choose Perform all rejection actions for this step AND all final rejection actions. (Final Rejection) so that offer letters rejected by your CFO are completely rejected.

Tip:

- Consider creating the following final approval actions:
  - Email alert to notify the user who submitted the offer letter request.
  - Field update to select the Offer Extended checkbox.
- Consider creating the following final rejection actions:
  - Email alert to notify the manager that the offer won't be extended.

Wrap Things Up

After creating the approval process, add the Approval History related list to the Candidates object page layout.

Tip: Consider adding the Items To Approve related list to your custom home page layouts. It gives users an instant view of the approval requests that are waiting for their response.

If available, use your sandbox to test the approval process, then activate it.

SEE ALSO:

- Define a Custom Object
- Set Up an Approval Process
- Prepare Your Org for Approvals

Approval History Reports

If you create a custom report type for approval process instances, users can view the historical details of completed and in-progress approval processes and their individual steps.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Fields Available for Approval History Reports
If you create a custom report type with Process Instance as the primary object and Process Instance Node as the related object, you can create approval history reports with various combinations of fields that enable you to view a detailed history of executed and in-progress approval processes and their individual steps.

Examples of Approval History Reports
See sample reports to learn how you can obtain approval history data.

Considerations for Approval History Reports
Understand the limitations and special behaviors when you create or view approval history reports, which provide a detailed history of approval processes and steps.

Fields Available for Approval History Reports

If you create a custom report type with Process Instance as the primary object and Process Instance Node as the related object, you can create approval history reports with various combinations of fields that enable you to view a detailed history of executed and in-progress approval processes and their individual steps.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Process Instance

A process instance represents one instance of an approval process. A new process instance is created each time a record is submitted for approval.

Field	Description
`Approval Process: Name`	Name of the approval process.
`Approval Process Instance ID`	ID of the approval process instance.

Field	Description
`Completed Date`	Date and time when the approval process instance was completed or recalled. If no step criteria are met and the record is auto-approved or auto-rejected, `Completed Date` and `Submitted Date` have the same values.
`Elapsed Days`	Length of time between when the record was submitted for approval and when the approval process was completed or recalled.
`Elapsed Hours`	
`Elapsed Minutes`	
`Last Actor: Full Name`	Full name of the user who most recently participated in the approval process instance. If no step criteria are met and the record is auto-approved or auto-rejected, `Last Actor: Full Name` and `Submitter: Full Name` have the same values.
`Object Type`	Object type of the record that was submitted for approval.
`Pending Step Name`	Name of the approval step at which the record is awaiting approval or rejection.
`Record Name`	Name of the record that was submitted for approval.
`Status`	Status of the approval process instance.
`Submitted Date`	Date and time when the record was submitted for approval.
`Submitter: Full Name`	Full name of the user who submitted the record for approval.

Process Instance Node

A process instance node represents an instance of an approval step. A new process instance node is created each time a record enters a step in an approval process. No process instance node is created when the record doesn't meet the step criteria or if the approval process instance is otherwise completed without entering the step.

Field	Description
`Step: Name`	Name of the approval step.
`Step: Completed Date`	Date and time when the approval step instance was completed or recalled.
`Step Elapsed Days`	Length of time between when the record entered the approval step and when the approval step instance was completed or recalled.
`Step Elapsed Hours`	
`Step Elapsed Minutes`	
`Step Last Actor: Full Name`	Full name of the user who most recently participated in the approval step instance.
`Step Start Date`	Date and time when the record entered the approval step.

Field	Description
`Step Status`	Status of the approval step instance.

SEE ALSO:

- Approval History Reports
- Considerations for Approval History Reports
- Examples of Approval History Reports

Examples of Approval History Reports

See sample reports to learn how you can obtain approval history data.

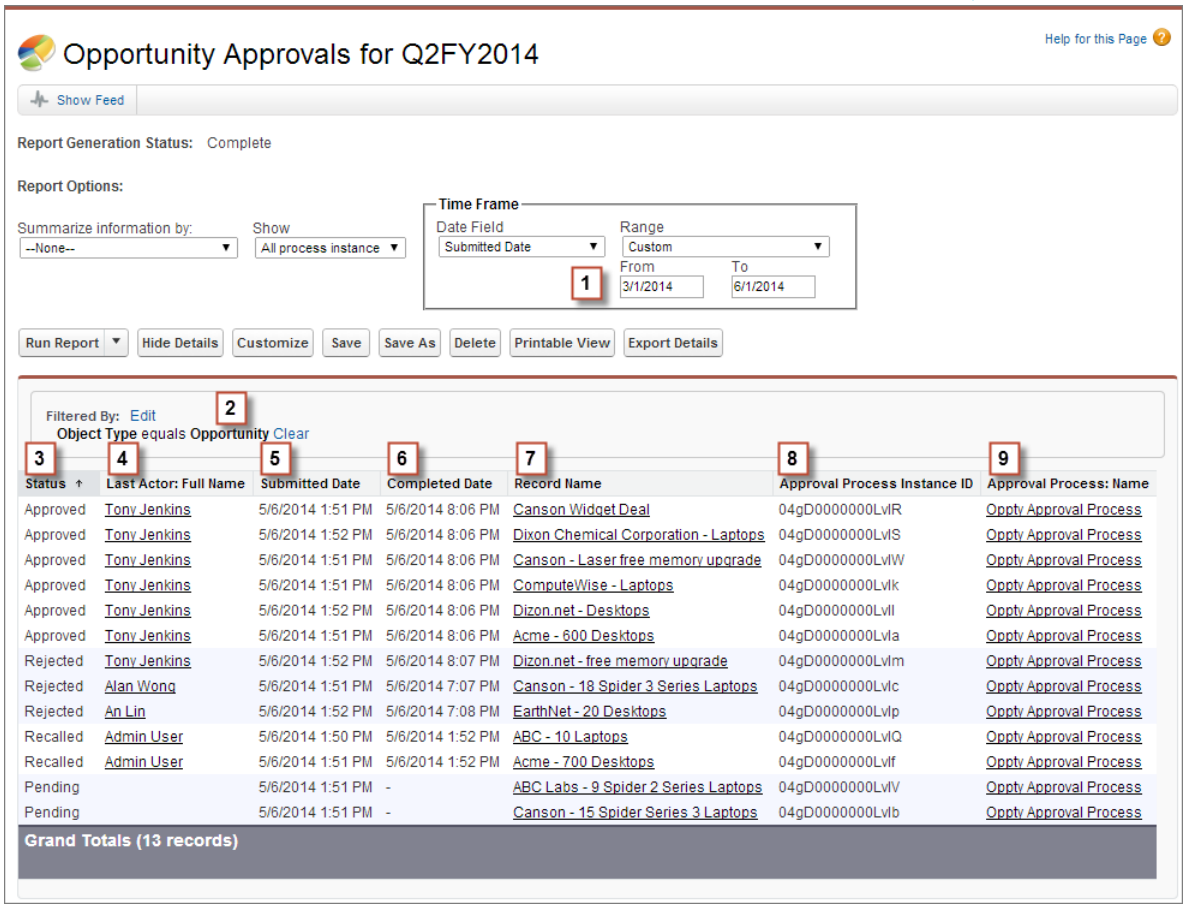
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

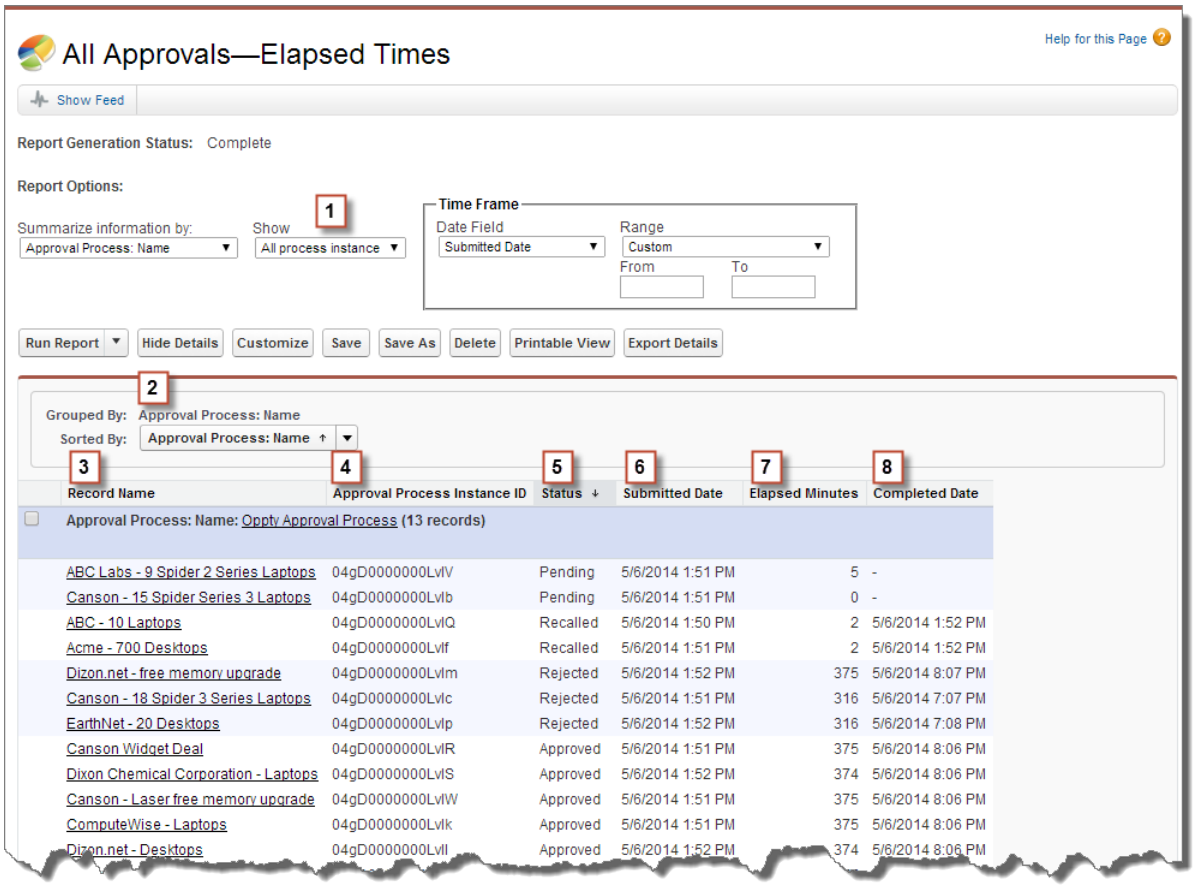
Report Example: Opportunity Approvals Submitted Within a Date Range

This sample report displays approval process instances that were submitted within a specified date range (1) for the Opportunity object (2). The results are sorted by status (3) and include the last actor (4), submitted date (5), completed date (6), record name (7), approval process instance ID (8), and approval process name (9).



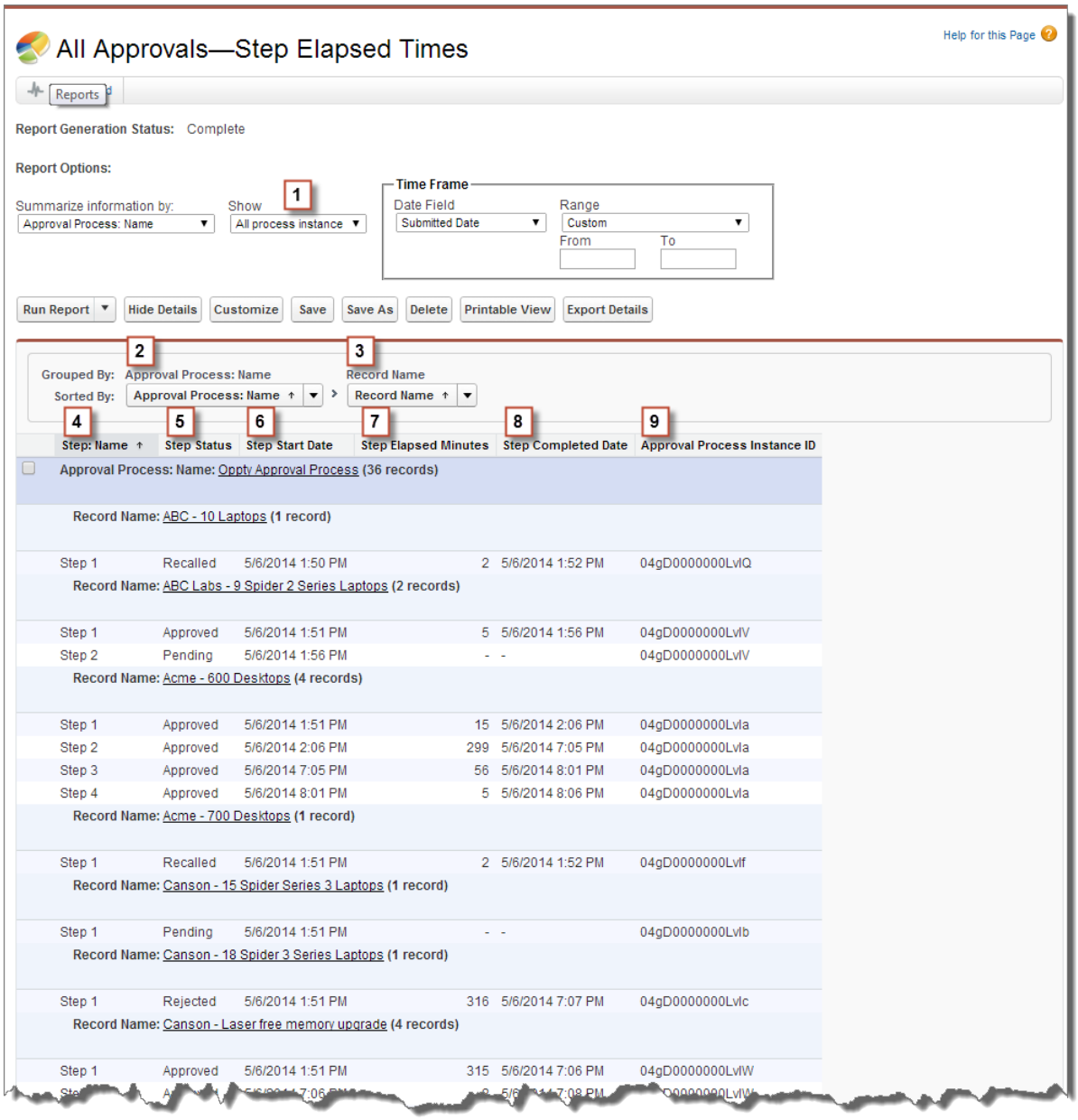
Report Example: Approvals—Elapsed Times

This sample report displays all approval process instances (1) and groups results by the approval process name (2). The results include the record name (3), approval process instance ID (4), status (5), submitted date (6), elapsed minutes (7), and completed date (8).



Report Example: Approval Steps—Elapsed Times

This sample report displays all approval process instances (1) and groups results by approval process name (2) and record name (3). The results are sorted by step name (4) and include step status (5), step start date (6), step elapsed minutes (7), step completed date (8), and approval process instance ID (9).



Notice that the previous sample report doesn't include the approvers for each step and the elapsed time for each approval request. To get this information, run a SOQL query by using the approval process instance ID from the report. The following sample SOQL query obtains the `ActorID` (user or queue that received the approval request) and the `ElapsedTimeInHours` (elapsed time since the approval request was sent) for the first pending step in the report.

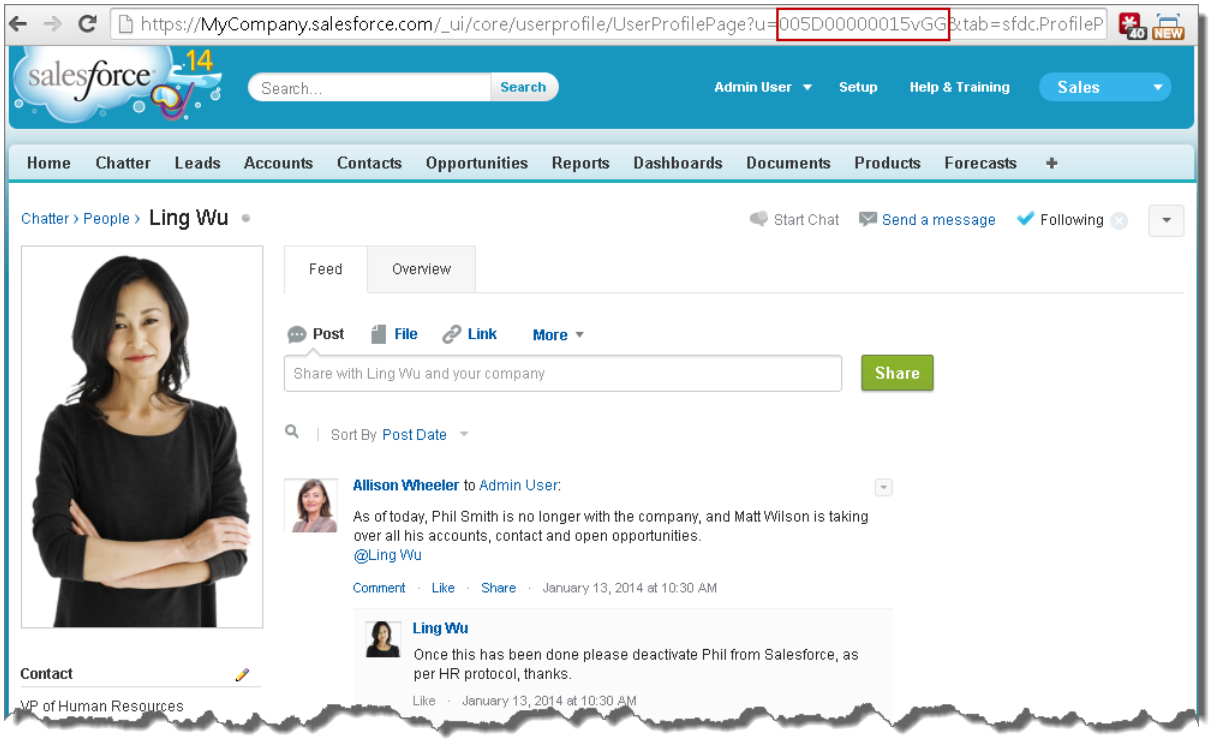
```
SELECT ActorId,ElapsedTimeInHours FROM ProcessInstanceWorkitem where processInstanceId =
'04gD0000000LvIV'
```

```
SELECT ActorId, ElapsedTimeInHours FROM ProcessInstanceWorkitem where processInstanceId = '04gD0000000LvIV'
```

Query Results - Total Rows: 1

ActorId	ElapsedTimeInHours
005D00000015vGGIAY	29.820833333333333

The sample query has only one result, and you can view that approver's user profile page by appending the resulting `ActorID` to the organization's base URL (`https://MyCompany.salesforce.com/005D00000015vGGIAY`), which gets redirected to the user profile page.



SEE ALSO:

- Approval History Reports

Considerations for Approval History Reports

Understand the limitations and special behaviors when you create or view approval history reports, which provide a detailed history of approval processes and steps.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Sandboxes available in: **Professional**, **Enterprise**, **Performance**, and **Unlimited** Editions

Considerations for Approval Processes That Were Completed Before or Pending During the Summer '14 Rollout

When Summer '14 became available for your organization, the approval history data was automatically populated for completed and pending approval processes. However, some approval history field values are never populated or are populated only after the approval process instance is next acted upon—such as when a user approves, rejects, or reassigns an approval request—after the Summer '14 rollout.

Object	When Fields are Populated
Process Instance	For approval process instances that were completed before the Summer '14 rollout, all Process Instance fields are automatically populated, with one exception: `Completed Date` is never populated for approval process instances that were completed before January 1, 2013. For approval process instances that were pending during the Summer '14 rollout, all Process Instance fields are automatically populated, with two exceptions: `Completed Date` and `Last Actor: Full Name` are populated only after the approval process instance is complete.
Process Instance Node	Never populated for approval process instances that were completed before the Summer '14 rollout. For approval process instances that were pending during the Summer '14 rollout, all Process Instance Node fields are populated only after the approval process instance is next acted upon after the Summer '14 rollout.

Additional exceptions apply to approval history data that is available only via SOQL queries of certain objects. See ProcessInstance, ProcessInstanceNode, ProcessInstanceStep, and ProcessInstanceWorkitem in the *Object Reference for Salesforce and Force.com*.

Considerations for the Sandbox Environment

If you copy approval history data to a sandbox, some field values are overwritten and don't reflect the actual approval history.

Object	Field	When an existing process instance or process instance node record is copied to a sandbox...
Process Instance	`Submitted Date`	This value is overwritten by the date and time when the process instance record is copied to the sandbox.
	`Submitter: Full Name`	This value is overwritten by the name of the user who copied the process instance record to the sandbox.

Object	Field	When an existing process instance or process instance node record is copied to a sandbox...
Process Instance Node	`Step Start Date`	This value is overwritten by the date and time when the process instance node record is copied to the sandbox.

SEE ALSO:

Fields Available for Approval History Reports

Approval History Reports

Manage Multiple Approval Requests

Transfer multiple approval requests from one user to another or remove multiple approval requests from the approval process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Transfer Pending Approval Requests

If users move to a new role before they complete all their pending approval requests, transfer the remainder to another user.

Remove Pending Approval Requests

If you want to clean up old approval requests—such as to delete an approval process—remove them from your Salesforce org. After approval requests are removed, the associated records are unlocked and removed from all approval processes, so they no longer appear on the approver's list of pending approval requests.

SEE ALSO:

Considerations for Managing Approvals

Transfer Pending Approval Requests

If users move to a new role before they complete all their pending approval requests, transfer the remainder to another user.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To transfer multiple approval requests:

- "Transfer Leads"

  AND

  "Transfer Record"

1. From Setup, enter *`Mass Transfer Approval Requests`* in the `Quick Find` box, then select **Mass Transfer Approval Requests**.
2. Search for the approval requests to transfer.
3. Select **Mass transfer outstanding approval requests to a new user**.
4. Look up and select the user to transfer the requests to.

   Make sure that the user can view the records that are associated with the approval requests.
5. Add comments.

   The comments you enter display on the Approval History related list.
6. Select each approval request that you want to transfer.

7. Click **Transfer**.

SEE ALSO:

Remove Pending Approval Requests
Considerations for Managing Approvals
Manage Multiple Approval Requests

Remove Pending Approval Requests

If you want to clean up old approval requests—such as to delete an approval process—remove them from your Salesforce org. After approval requests are removed, the associated records are unlocked and removed from all approval processes, so they no longer appear on the approver's list of pending approval requests.

1. From Setup, enter `Mass Transfer Approval Requests` in the `Quick Find` box, then select **Mass Transfer Approval Requests**.
2. Search for the approval requests that you want to remove.
3. Select **Mass remove records from an approval process**.
4. Add comments.

   The comments you enter display on the Approval History related list.
5. Select each approval request to remove from the approval process.
6. Click **Remove**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To remove multiple approval requests:

- "Transfer Leads"

  AND

  "Transfer Record"

SEE ALSO:

Transfer Pending Approval Requests
Considerations for Managing Approvals
Manage Multiple Approval Requests

Approval Requests for Users

Your admin can set up approval processes that let you and other users submit records for approval, which results in *approval requests*.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Submit a Record for Approval
Depending on your org's customizations, you can submit a record for approval directly from that record.

Withdraw an Approval Request
If you submitted a record for approval but suddenly need to update information in the record, recall the approval request. However, whether you can recall an approval request depends on how your admin configured the approval process that the record was submitted to.

Respond to an Approval Request

When you receive an approval request, respond to it by approving, rejecting, or reassigning it. Depending on which Salesforce experience you're using, you have different options. Approval request comments are limited to 4,000 characters. In Chinese, Japanese, or Korean, the limit is 1,333 characters.

What Does This Approvals Error Mean?

Here are some errors that you could encounter when submitting a record for approval or responding to an approval request.

Approval History Status

To track where a record is in an approval process, view its Approval History related list.

Approval User Preferences

Identify a delegated approver and control whether you receive approval request emails.

Submit a Record for Approval

Depending on your org's customizations, you can submit a record for approval directly from that record.

1. Go to the record that you want to submit for approval.
2. Make sure it's ready to be submitted.

   Before you can submit a record for approval, it must meet the criteria for an active approval process. If you're not sure what the requirements are, ask your admin.
3. Click **Submit for Approval**.

   If an approval process applies to the record, Salesforce begins the approval process. This button isn't available after the record has been submitted.

To keep tabs on the progress of your submitted approval, we recommend following the approval record in Chatter.

SEE ALSO:

Withdraw an Approval Request

Approval User Preferences

Approval Requests for Users

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To submit a record for approval:

- "Read" on the record

Withdraw an Approval Request

If you submitted a record for approval but suddenly need to update information in the record, recall the approval request. However, whether you can recall an approval request depends on how your admin configured the approval process that the record was submitted to.

1. Go to the detail page for the record associated with the approval request.
2. In the Approval History related list, click **Recall Approval Request**.

SEE ALSO:

- Submit a Record for Approval
- Approval User Preferences
- Approval Requests for Users

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To withdraw an approval request:

- "Read" on the record

Respond to an Approval Request

When you receive an approval request, respond to it by approving, rejecting, or reassigning it. Depending on which Salesforce experience you're using, you have different options. Approval request comments are limited to 4,000 characters. In Chinese, Japanese, or Korean, the limit is 1,333 characters.

Respond from...	Lightning Experience	Salesforce Classic	Salesforce1
An in-app notification	✅		✅
An email notification	✅	✅	✅
The record	✅	✅	✅
Chatter	✅	✅	✅
Home		✅	

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To respond to an approval request from within Salesforce:

- "Read" on the associated record

To respond to an approval request from an email:

- "API Enabled"

In-App Notification

Depends on the `Receive Approval Request Emails` field in your approver preferences. If notifications are enabled for your org, you receive a notification whenever you receive an approval request email.

- Respond from the notification if your admin enabled actionable notifications.
- To open the approval request, click the notification.

Email Notification

Depends on the `Receive Approval Request Emails` field in your approver preferences.

- To open the approval request, click the link in the email.
- Reply to the email if your admin enabled email approval response.

Record

Respond from the Approval History related list.

Chatter

Depends on if your admin has enabled Approvals in Chatter and you haven't opted out of receiving approval requests through Chatter posts.

- Respond from the post if your admin enabled actionable notifications.
- Click the name of the record, then respond from the Approval History related list.

Home

Depends on if your admin added the Items to Approve component to your home page. From the Home tab, respond from the Items to Approve component.

Tip: From this component, you can respond to multiple requests at once.

IN THIS SECTION:

Respond to an Approval Request by Replying to the Email Notification

If your admin has enabled email approval response, you can approve or reject requests by responding to the email notification, regardless of which Salesforce experience or mobile email client you're using. Delegated approvers can also respond to approval requests by email.

Troubleshoot Email Responses to Approval Requests

When email responses aren't working correctly, review these common issues.

SEE ALSO:

Approval User Preferences

Approval Requests for Users

Respond to an Approval Request by Replying to the Email Notification

If your admin has enabled email approval response, you can approve or reject requests by responding to the email notification, regardless of which Salesforce experience or mobile email client you're using. Delegated approvers can also respond to approval requests by email.

Email approval response works in all languages that Salesforce supports. The response word or phrase is checked using the current user language dictionary. If no matches are found, the response word or phrase is checked in all other language dictionaries.

1. In the first line of your reply to the email notification, enter one of the supported response words.

   Periods and exclamation marks are allowed at the end of the word.

Approval Words	Rejection Words
approve	reject
approved	rejected
yes	no

2. Optionally, in the second line of your reply, add comments.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To respond to an approval request via email:

- "API Enabled"

3. Send the email.

SEE ALSO:

Approval User Preferences

Approval Requests for Users

Troubleshoot Email Responses to Approval Requests

When email responses aren't working correctly, review these common issues.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

I'm not receiving approval requests by email.

Here are a few possible reasons why.

- Your approval preferences opt you out of approval request emails.
- Your mail server thinks the approval request email is spam. Contact your email admin, who can check the logs of all inbound email to see if it's being delivered, rejected, or marked as spam.
- Your email admin needs to whitelist the Salesforce email addresses that the approval requests come from.
- Email delivery time can vary based on your ISP or connection.

My response wasn't delivered.

- An email approval request can only be processed once. If another user has responded to the approval request before you do, you get an error.
- You must have the "API Enabled" user permission to respond to approval requests by email.

I received an email that said, "`The word used to approve or reject the item was not understood.`"

Salesforce doesn't process replies to error emails. Reply again to the original email notification, but this time use one of the supported response words on page 456.

I received an email that said, "`You are not authorized to update the referenced object.`"

The approval request email is tied to your email address. You receive this error if you forward the request to another email address or if your email client lets you respond from multiple email addresses. Reply again to the original email notification, but this time reply from the same email address that received the email approval request.

SEE ALSO:

What Does This Approvals Error Mean?

Respond to an Approval Request by Replying to the Email Notification

Approval Requests for Users

What Does This Approvals Error Mean?

Here are some errors that you could encounter when submitting a record for approval or responding to an approval request.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Manager undefined

This approval request requires the next approver to be determined by the `Field Name` field. This value is empty.

Salesforce tried to route the approval request based on a hierarchical field, such as `Manager`. However, the field has no value or specifies an inactive user. This error can occur when a user submits a record for approval or when an approver responds to an approval request.

Required fields are missing: [`FieldName`].

The approval process includes a field update that fails standard validation rules for the identified field. This error can occur even if the field isn't visible on your page layout.

Note: Salesforce doesn't make sure that field updates pass *custom* validation rules on fields.

SEE ALSO:

Troubleshoot Email Responses to Approval Requests

Approval Requests for Users

Approval History Status

To track where a record is in an approval process, view its Approval History related list.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Status	Definition
Submitted	The record has been submitted for approval.
Pending	The record has been submitted for approval and is awaiting approval or rejection.
Approved	The record has been approved.
Rejected	The record has been rejected.
Reassigned	The record has been submitted for approval but assigned to a different approver.
Recalled	The record was submitted for approval but recalled from the approval process.

SEE ALSO:

Approvals

Submit a Record for Approval

Respond to an Approval Request

Approval Requests for Users

Approval User Preferences

Identify a delegated approver and control whether you receive approval request emails.

From your personal settings, enter *Approver Settings* in the Quick Find box, then select **Approver Settings**. No results? Enter *Personal Information* in the Quick Find box, then select **Personal Information**.

Field	Description
Delegated Approver	Your alternate approver. If populated, this user receives the same approval requests as you do. Delegated approvers can't reassign approval requests; they can only approve or reject approval requests.
Manager	Depending on how your admin sets up approval processes, requests for approval can automatically be routed to your manager.
Receive Approval Request Emails	Controls whether you receive approval request notifications by email, in Salesforce1, or in Lightning Experience. If you select **Never**, you don't receive approval request notifications. However, you still receive approval request emails from a queue, depending on how your admin configured the queue email.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Opt Out of Approval Request Posts in Chatter

By default, when your org enables Approvals in Chatter, you are notified about approval requests by both email and a Chatter post. To stop receiving the posts in Chatter, opt out. If you do opt out, the posts don't appear in your feed but they do appear in the associated record's feed.

What Happens When You Opt Out of Chatter Approval Requests?

By default, when your org has Approvals in Chatter enabled, you receive approval request notifications by email and Chatter. Here's what happens when you opt out of the Chatter posts.

SEE ALSO:

Approval Requests for Users

Opt Out of Approval Request Posts in Chatter

By default, when your org enables Approvals in Chatter, you are notified about approval requests by both email and a Chatter post. To stop receiving the posts in Chatter, opt out. If you do opt out, the posts don't appear in your feed but they do appear in the associated record's feed.

1. From your personal settings, enter *My Feeds* in the Quick Find box, then select **My Feeds**.
2. Deselect the option to receive approval requests as posts.
3. Save your changes.

SEE ALSO:

Approval User Preferences

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view an approval request post for a record:

- "Read" on the record

What Happens When You Opt Out of Chatter Approval Requests?

By default, when your org has Approvals in Chatter enabled, you receive approval request notifications by email and Chatter. Here's what happens when you opt out of the Chatter posts.

- If you opt out while an approval that you're assigned to is in progress, you see new notification posts only if you're following the approval record.
- If you're following the approval record, you see approval posts from the record with non-approver content.
- In the approval notification posts that you've already received, you see non-approver content.
- The Approve and Reject buttons disappear from existing posts in your feed.

SEE ALSO:

Approval User Preferences

Opt Out of Approval Request Posts in Chatter

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Approval Process Terminology

The following terminology is used for approval processes in Salesforce.

Approval Actions
An approval action occurs when a step is approved by all the required approvers.

Approval Process
An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including who to request approval from and what to do at each point of the process.

Approval Request
An approval request is an email, Salesforce1 notification, Lightning Experience notification, or Chatter post notifying the recipient that a record was submitted for approval and that his or her approval is requested.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Approval Steps
Approval steps define the chain of approval for a particular approval process. Each step determines which records can advance to that step, who to assign approval requests to, and whether to let each approver's delegate respond to the requests. The first step specifies what to do if a record doesn't advance to that step. Later steps specify what happens if an approver rejects the request.

Assigned Approver
The assigned approver is the user responsible for responding to an approval request.

Delegated Approver
A delegated approver is a user appointed by an assigned approver as an alternate for approval requests.

Email Approval Response
Email approval response lets users respond to approval requests by replying to an email notification.

Initial Submission Actions
An initial submission action occurs when a user first submits a record for approval. By default, the record is locked.

Final Approval Actions
Final approval actions occur when all required approvals have been given for a record.

Final Rejection Actions
A final rejection action occurs when an approver rejects the request and it goes to the final rejection state.

Outbound Message
An outbound message sends information to a designated endpoint, like an external service. Outbound messages are configured from Setup. You must configure the external endpoint and create a listener for the messages using the SOAP API.

Process Instance
A process instance represents one instance of an approval process. A new process instance is created each time a record is submitted for approval.

Process Instance Node
A process instance node represents an instance of an approval step. A new process instance node is created each time a record enters a step in an approval process. No process instance node is created when the record doesn't meet the step criteria or if the approval process instance is otherwise completed without entering the step.

Recall Actions
A recall action occurs when a submitted approval request is recalled. By default, the record is unlocked.

Record Locking
Record locking prevents users from editing a record, regardless of field-level security or sharing settings. By default, Salesforce locks records that are pending approval. Only admins can edit locked records.

SEE ALSO:

Approvals
Automated Actions
Set Up an Approval Process

Visual Workflow

Visual Workflow lets you automate business processes by building flows and distributing them to the right users or systems. A *flow* is an application that can execute logic, interact with the Salesforce database, call Apex classes, and collect data from users. You can build flows by using the Cloud Flow Designer.

Flows can either require user interaction—perhaps a wizard or guided UI for data entry—or run in the background on their own—perhaps something that automatically transfers records when a user's role changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Limits and Considerations for Visual Workflow

When designing, managing, and running flows, consider the permissions, use limits, and data issues.

Create a Flow

Once you understand the process that you want to automate, design a flow in the Cloud Flow Designer for that process.

Manage Your Flows

Use the flow detail page to do anything with your flow outside of designing it—such as activating a flow, testing it, or viewing its properties.

Distribute Your Flow

Once you've created and activated a flow version, distribute it to users so that they can run it. The right distribution method depends on the users that you want to distribute the flow to: internal users, external users, systems, or other organizations.

Why Did My Flow Interview Fail?

To troubleshoot a failed flow interview, use the flow fault email. You can also set up temporary Screen or Send Email elements to identify the problem.

Flow Interviews

A *flow interview* is a running instance of a flow. A *flow* is an application built by your administrator that asks you for inputs and does something in Salesforce based on those inputs.

Visual Workflow Terminology

SEE ALSO:

Which Automation Tool Do I Use?

Flow Building Blocks

What's the Difference Between Workflow and Visual Workflow?

Despite their similar names, Workflow and Visual Workflow are separate Salesforce features for automating business processes.

Workflow

Workflow enables you to set up *workflow rules*. A workflow rule identifies what kinds of record changes or additions trigger specified *workflow actions*, such as sending email alerts and updating record fields.

Workflow rules and actions are associated with a specific object (and can cross objects only to update fields on a related master record).

Visual Workflow

Visual Workflow enables you to create *flows*, which are triggered by users rather than events. Unlike Workflow, which always executes rules and actions behind the scenes, Visual Workflow offers screens for displaying and collecting information from the user running the flow.

Note: You can, however, participate in a pilot program that enables workflow rules to execute flows behind the scenes. The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Flows aren't tied to any one object. They can look up, create, update, and delete records for multiple objects.

In case you're wondering, the "visual" in Visual Workflow refers to the ability to visually build flows using the Cloud Flow Designer. Its drag-and-drop user interface lets you build flows without writing any code.

SEE ALSO:

Which Automation Tool Do I Use?

Workflow

Visual Workflow

Limits and Considerations for Visual Workflow

When designing, managing, and running flows, consider the permissions, use limits, and data issues.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Limits for Visual Workflow

When using Visual Workflow, keep flow limits and Apex governor limits in mind.

Flow Best Practices

Before you begin building and distributing flows, understand the best practices.

Considerations for Designing Flows

When you design flows, keep certain guidelines in mind.

Considerations for Managing Flows

When managing flows, consider the administration and activation limits.

Considerations for Running Flows

When you run or test a flow, keep certain limits and guidelines in mind.

Flow Accessibility

Visual Workflow is 508-compliant with a few exceptions.

SEE ALSO:

Cloud Flow Designer

Considerations and Limitations for Flows in Lightning Pages (Beta)

Limits for Visual Workflow

When using Visual Workflow, keep flow limits and Apex governor limits in mind.

Maximum number of versions per flow	50
Maximum number of executed elements at run time	2,000
Maximum number of active flows and processes per org	500
Maximum number of flows and processes per org	1,000
Maximum number of flow interviews or groups of scheduled actions (from processes) that are waiting at one time	30,000
Maximum number of flow interviews that are resumed or groups of scheduled actions that are executed per hour	1,000
Maximum number of relative time alarms defined in flow versions or schedules based on a field value in processes	20,000

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Apex Governor Limits that Affect Flows

Salesforce strictly enforces limits to ensure that any runaway flows don't monopolize shared resources in the multitenant environment. Per-transaction limits, which Apex enforces, govern flows. If an element causes the transaction to exceed governor limits, the system rolls back the entire transaction. The transaction rolls back even if the element has a defined fault connector path.

Flows in Transactions

Each flow interview runs in the context of a *transaction*. A transaction represents a set of operations that are executed as a single unit. For example, a transaction can execute Apex triggers and escalation rules in addition to a flow interview. If one interview in a transaction fails, all the interviews in the transaction are rolled back, as well as anything else the transaction did. The transaction doesn't retry any of the operations—including the flow interview.

Flow Bulkification in Transactions

Programmers can design their code so that similar actions are performed together in one batch. For example, one operation to create 50 records rather than 50 separate operations that each create one record. This process is called *bulkification*, and it helps your transaction avoid governor limits. If you're working with flows, you don't even have to think about bulkification. Flow interviews bulkify actions for you automatically.

SEE ALSO:

Visual Workflow

Limits and Considerations for Visual Workflow

Apex Governor Limits that Affect Flows

Salesforce strictly enforces limits to ensure that any runaway flows don't monopolize shared resources in the multitenant environment. Per-transaction limits, which Apex enforces, govern flows. If an element causes the transaction to exceed governor limits, the system rolls back the entire transaction. The transaction rolls back even if the element has a defined fault connector path.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Description	Per-Transaction Limit[1]
Total number of SOQL queries issued (Record Update, Record Delete, Record Lookup, and Fast Lookup element executions)	100
Total number of records retrieved by SOQL queries (across all Record Update, Record Delete, Record Lookup, and Fast Lookup elements executed in all interviews in the transaction)	50,000
Total number of DML statements issued (Record Create, Record Update, Record Delete, Fast Create, Fast Update, and Fast Delete executions)	150
Total number of records processed as a result of DML statements	10,000

[1] Autolaunched flows are part of the larger transaction through which they were launched. For example, flows launched from a process are executed with the process actions as part of the larger transaction. Flows with Screen elements can span multiple transactions. A new transaction begins each time the user clicks **Next** in a screen. Flows with Wait elements span multiple transactions. A transaction ends when a flow interview begins to wait for an event. When the flow interview resumes, a new transaction begins. Everything after the Wait element is executed as part of a batch transaction that includes other resumed interviews.

SEE ALSO:

Apex Developer Guide: Execution Governors and Limits

Limits for Visual Workflow

Flows in Transactions

Each flow interview runs in the context of a *transaction*. A transaction represents a set of operations that are executed as a single unit. For example, a transaction can execute Apex triggers and escalation rules in addition to a flow interview. If one interview in a transaction fails, all the interviews in the transaction are rolled back, as well as anything else the transaction did. The transaction doesn't retry any of the operations—including the flow interview.

In each transaction, Salesforce enforces governor limits to prevent shared resources from being depleted. Because multiple Salesforce organizations share the same resources, Salesforce prevents one organization from depleting all the resources and leaving the other organizations high and dry. It's similar to an apartment building that uses one cache of water to service every tenant. If your neighbor

uses all the water, you can't take a shower. (It's trite, but hopefully you get the idea.) Per-transaction governor limits help prevent such things from happening.

IN THIS SECTION:

When Does a Flow's Transaction Start?

Depending on how the flow was distributed, a transaction that runs an interview for that flow starts in different ways.

When Does a Flow's Transaction End?

When a transaction ends depends on whether the flow contains certain elements and whether it originally started because a record was changed.

SEE ALSO:

Flow Bulkification in Transactions

When Does a Flow's Transaction Start?

Depending on how the flow was distributed, a transaction that runs an interview for that flow starts in different ways.

Distribution Method	**Transaction starts when...**
Process Builder[1]	A record is created or updated.
Flow URL	The URL is accessed.
Custom button or link	The button or link is clicked.
Visualforce page	The page is accessed.
`Interview.start()` method	If the method starts via a `before` or `after` trigger, the transaction starts when a record is created or updated. Otherwise, the transaction starts when the method (or a parent method) is invoked. The `start()` method shares its limits with other operations in the transaction and other methods in the class.
REST API (Custom Actions or Flows resource)	When the REST call is made. Depending on how the REST call is implemented, the limits can be shared with other operations.

[1]The same also applies if the flow is distributed through a workflow rule. The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Note: When a Screen or Wait element is executed, the existing transaction ends and a new one begins.

When Does a Flow's Transaction End?

When a transaction ends depends on whether the flow contains certain elements and whether it originally started because a record was changed.

The transaction ends when:

- A Screen or Wait element is executed
- The order of execution has completed—if the flow was triggered when a record was created or updated
- All the interviews in the transaction have finished

Tip: If you think that a flow's interview is likely to hit governor limits within its transaction, consider adding a Wait element or a Screen element.

If the interview is one of many things being done in a given transaction, that interview shares the transaction's governor limits with the other operations.

Example: You update 100 cases through Data Loader. Due to the order of execution in a transaction and the customizations in your organization, here's what happens.

	Transaction Operation	DML Statement Used	SOQL Query Used
1	Cases are saved to the database, but aren't committed yet.		
2	Case assignment rules are executed. Each case's owner is updated.	✅	
3	Case escalation rules are executed. If any case has been open for 10 days, an email is sent to the owner.		
4	Process is started.		
5	Process looks up the case's account.		✅
6	If the account is hot, process uses Chatter to notify the account owner that there's a new case associated with the account.	✅	
7	Process launches a flow interview.		
8	Flow interview looks up the parent account and how many cases it has.		✅
9	Flow interview checks whether the account has more than five open cases.		
10	If it does, flow interview looks up the account's division manager then posts on the account's Chatter feed to notify the division manager and account owner.	✅	✅
11	If it doesn't, flow interview posts on the account's Chatter feed to notify only the account owner.	✅	

SEE ALSO:

Apex Developer Guide: Triggers and Order of Execution

Flow Bulkification in Transactions

Programmers can design their code so that similar actions are performed together in one batch. For example, one operation to create 50 records rather than 50 separate operations that each create one record. This process is called *bulkification*, and it helps your transaction avoid governor limits. If you're working with flows, you don't even have to think about bulkification. Flow interviews bulkify actions for you automatically.

IN THIS SECTION:

SEE ALSO:

Flows in Transactions

How Does Flow Bulkification Work?

Interview operations are bulkified only when they execute the same element. That means that the interviews must all be associated with the same flow.

When multiple interviews for the same flow run in one transaction, each interview runs until it reaches a bulkifiable element. Salesforce takes all the interviews that stopped at the same element and intelligently executes those operations together. If other interviews are at a different element, Salesforce then intelligently executes those operations together. Salesforce repeats this process until all the interviews finish.

If, despite the bulkification, any interview hits a governor limit, all the interviews in the transaction fail. Any operations that the interviews performed are rolled back, and the transaction doesn't try to perform the operations again.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: When you upload 100 cases, the flow MyFlow_2 triggers one interview for each case.

- 50 interviews stop at Record Create element **Create_Task_1**.
- The other 50 interviews stop at Record Create element **Create_Task_2**.

The result? At least two groups of bulk operations to execute.

- One for the 50 interviews that execute **Create_Task_1**
- One for the 50 interviews that execute **Create_Task_2**

Which Flow Elements Can Be Bulkified?

Flows can bulkify any element that performs a DML statement or SOQL query or does something else external to the flow, like sending an email.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Elements that create, update or delete records

When a record is created, updated, or deleted, the transaction performs a DML statement.

- Create elements (Record Create, Fast Create)
- Update elements (Record Update, Fast Update)
- Delete elements (Record Delete, Fast Delete)
- Quick Action elements
- Post to Chatter elements

- Submit for Approval elements
- Apex elements—depending on your organization (invocable Apex only)

Elements that look up records

When fields on a record are looked up, the transaction performs a SOQL query.

- Lookup elements (Record Lookup, Fast Lookup)
- Record Update elements
- Record Delete elements
- Apex elements—depending on your organization (invocable Apex only)

Elements that send emails

- Send Email elements
- Email Alert elements
- Apex elements—depending on your organization (invocable Apex only)

Note:

- Unlike invocable Apex, Apex Plug-in elements aren't bulkified.
- Although invocable Apex is bulkified, the flow has no way of knowing what the invoked methods' operations are. If you want those operations to also be bulkified, make sure the code follows bulkification best practices.

SEE ALSO:

Apex Developer Guide: Running Apex within Governor Execution Limits

Example of Flow Bulkification

This example demonstrates how operations are bulkified for a flow when 100 cases are updated through Data Loader.

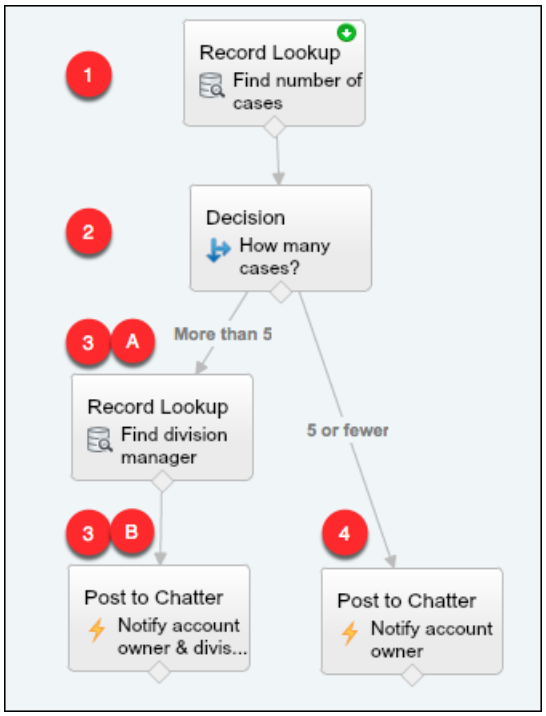
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The Associated Flow

You'll understand the concepts better if you understand the design of the associated flow.



The flow:

1. Looks up the case's parent account and how many open cases that account has.
2. Checks whether the account has more than five cases open.
3. If the account has more than five open cases:
   a. Looks up the division manager for the account.
   b. Posts on the account's Chatter feed to notify the division manager and the account owner.
4. If the account has five or fewer open cases, posts on the account's Chatter feed to notify only the account owner.

The Bulkified Interviews

When you update the records, one flow interview is created for each case simultaneously. All of the interviews are associated with the same flow. Each interview runs until it gets to a bulkifiable element.

The first interview goes through the Record Lookup element **(1)**. Because Record Lookups can be bulkified, the interview waits there until all the other interviews have done the same. Then, Salesforce executes all the Record Lookup operations together (because they're all for the same element in the same flow). Instead of 100 SOQL queries, the transaction issues one SOQL query.

The first interview is evaluated by the Decision element **(2)**. The account has six cases, so the interview is routed down the "More than 5" path. The interview proceeds to the second Record Lookup element **(3a)**. Because it's a bulkifiable element, the interview waits there.

The second interview is evaluated by the Decision element **(2)**. This account has one case, so the interview is routed down the "5 or fewer" path. The interview proceeds to the Post to Chatter element **(4)**. This element is also bulkifiable, so the interview waits there.

After all the interviews have been processed, 30 are waiting to execute the second Record Lookup element **(3a)** and the remaining 70 are waiting to execute the Post to Chatter element **(4)**.

Salesforce executes all the Record Lookup **(3a)** operations for the first 30 interviews together. Instead of 30 separate SOQL queries, the transaction issues one.

Next, the transaction returns to the Post to Chatter element **(4)**, where the 70 interviews are ready to execute their Post to Chatter operations. Remember, these are the interviews whose accounts don't have more than five cases. Salesforce executes the Post to Chatter operations together. Instead of 100 separate DML statements to create each Chatter post, the transaction issues one DML statement to create all 100 posts at one time. Because the Post to Chatter element isn't connected to a subsequent element, those 70 interviews finish.

The 30 interviews—which looked up the relevant division manager—proceed to the final Post to Chatter element **(3b)**. When all 30 interviews are ready, Salesforce executes all 30 Post to Chatter operations together. Instead of issuing 30 separate DML statements for the individual Chatter posts, it issues one. Because the Post to Chatter element isn't connected to another element, those 30 interviews finish.

Flow Best Practices

Before you begin building and distributing flows, understand the best practices.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Plan out your flow before you start building.

Write or draw out all the details of your business process. That way, you have a clear idea of what information you need, where you're getting that information from, and what logic and actions to perform. Doing so makes building the corresponding flow much easier.

Build your flows in a test environment—like a sandbox or Developer Edition org.

The last thing you want to do is accidentally change records in your company's production org. Build your flows in a separate environment. That way, you can enter fake data and test various permutations of your flow without worrying about changing or deleting data that your users actually need.

Never hard-code Salesforce IDs.

IDs are org-specific, so don't hard-code new or existing IDs. Instead, let Salesforce create the IDs, and pass them into variables when the flow starts. You can do so, for example, by using merge fields in URL parameters or by using a lookup element.

Wait until the end of the flow to make changes to the database.

Have you heard about flow limits? Because flows operate under Apex governor limits, the sky is not the limit. To avoid hitting those limits, we recommend bunching all your database changes together at the end of the flow, whether those changes create, update, or delete records.

Control when running users can navigate backward.

If the flow commits changes to the database between two screens, don't let users navigate from the later screen to the previous screen. Otherwise, the flow can make duplicate changes to the database.

Provide an error handler.

Sad to say, but sometimes a flow doesn't perform an operation that you configured it to do. Perhaps the flow is missing crucial information, or the running user doesn't have the required permissions. By default, the flow shows an error message to the user and emails the admin who created the flow. However, you can control that behavior. See Customize What Happens When a Flow Fails for more information and recommendations.

Save early and often.

Sometimes the Cloud Flow Designer falls victim to unexpected problems, like losing Internet access. Salesforce doesn't save your changes automatically, so it's up to you to save your work. Save as often as possible, so that you don't accidentally lose a few hours' worth of work.

Test as many permutations of your flow as you possibly can.

As with all customizations in Salesforce, it's important to test your work. This is especially true if your flow uses branching or other complex logic. Make sure that you test as many possibilities as you can think of before you distribute the flow to your users.

Considerations for Designing Flows

When you design flows, keep certain guidelines in mind.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Deleting Variables

If you delete an sObject variable or sObject collection variable, any variable assignments that use the deleted variable are set to `null`.

Manipulating Percentage Values

Test your flows carefully if they use sObject variables to manipulate percentage values. When you insert a value into an sObject variable's percentage field and then reference that field in a formula, the value is automatically divided by 100.

For example, an opportunity's Probability field is set to 100. If you assign that value to sObject variable `{!Opportunity.Probability}`, the value is still 100. But if you create a formula whose expression is `{!Opportunity.Probability}`, the value is 1.

Referring to Blank Fields or Resources

If you leave any field or resource value blank, that value is `null` at run time. To treat a text value as an empty string instead of `null`, set it to `{!$GlobalConstant.EmptyString}`.

Boolean Types Treat `null` Differently than `false`

Flow treats `null` as a different value than `false`. For example, if you try to find a record whose checkbox field is set to `null`, no records are returned. Instead, look for records where the checkbox field is set to `false`. If you're using a variable (such as `myCheckbox = {!varBoolean}`), make sure that the variable isn't set to null before you reference it in your record filter or condition.

Setting the Record Type

To set the record type for a record, use the ID of the record type. Look up the record type by its name and then store its ID in the flow.

For example, use a Record Lookup element to find the RecordType record whose Name is "Reduction Order". Then store that record type's ID in a variable. You can then use the variable to set the `Order Record Type` field on an order record.

Working with Person Accounts

If your org uses person accounts, reference `Contact.Salutation` instead of `Account.Salutation`.

External Objects

External objects aren't supported in flows.

IN THIS SECTION:

Considerations for the Cloud Flow Designer

When you create a flow in the Cloud Flow Designer, familiarize yourself with its limitations and behaviors. For example, it supports a handful of locales and can't open flows from managed packages.

Guidelines for Working with Large Flows

Business processes can be complex. When your flow is too large for the canvas, control the zoom, search in the Explorer tab, or collapse the left side panel.

Considerations for Two-Column Flows (Beta)

If your org has Lightning runtime enabled, you can control whether a flow displays in one column or two columns when you distribute that flow. Before you use this beta feature, though, understand how flow layout currently behaves.

Limitations for Multi-Select Choice Fields

Multi-select checkboxes and multi-select picklist fields let flow users select multiple choices in a screen field. Before you start using multi-select choice fields, understand how they work in flows, both when you design the flows and when your users run them.

Limitations for Flow Formulas

When you create a formula resource or add validation to a screen input field, understand the formula limitations in Visual Workflow.

Limitations for Time-Based Flows

Before you design flows that contain one or more Wait elements, understand the limitations and guidelines.

SEE ALSO:

Create a Flow

Flow Operators

Limits and Considerations for Visual Workflow

Cross-Object Field References in Flows

Considerations for the Cloud Flow Designer

When you create a flow in the Cloud Flow Designer, familiarize yourself with its limitations and behaviors. For example, it supports a handful of locales and can't open flows from managed packages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- At run time, time zones for date/time values can differ from what you see in the Cloud Flow Designer. During run time, date/time values reflect the running user's time zone settings in Salesforce. In the Cloud Flow Designer, date/time values reflect the time zone set on your computer. The Cloud Flow Designer appends the GMT offset to your date/time value.
- The Cloud Flow Designer doesn't support UTF-8 encoding for text in user input fields.
- The Cloud Flow Designer contains embedded fonts for all locales it supports. The supported locales are:
  - English (US)
  - French (France)
  - German (Germany)
  - Spanish (Spain)
  - Japanese (Japan)
  - Chinese (Traditional)
  - Chinese (Simplified)

  If you enter unsupported characters for a supported locale, they're displayed using system fonts instead of the embedded fonts.

  In unsupported locales, your system font settings are used to display all characters in the Cloud Flow Designer.
- The Cloud Flow Designer can't open flows that are installed from managed packages.
- Don't enter the string `null` as the value of a text field in the Cloud Flow Designer.
- The Cloud Flow Designer has access to information that exists when you open it. If you modify data or metadata in your organization and need to refer to it in a flow, close and reopen the Cloud Flow Designer. For example, if you add a custom field or modify an Apex class with the Cloud Flow Designer open, close and reopen the Cloud Flow Designer.
- The Cloud Flow Designer uses the permissions and locale assigned to the current user.
- If you open a flow that was last opened in Winter '12 or earlier, each Boolean decision is converted to a multi-outcome Decision element that:
  - Uses the same name as the old decision.
  - Takes the unique name of the old decision, appended with "_switch".

- Has an outcome labeled “True”. This outcome’s unique name matches that of the old decision, and its conditions are migrated from the True outcome of the old decision.
- Has a default outcome labeled “False”.

Guidelines for Working with Large Flows

Business processes can be complex. When your flow is too large for the canvas, control the zoom, search in the Explorer tab, or collapse the left side panel.

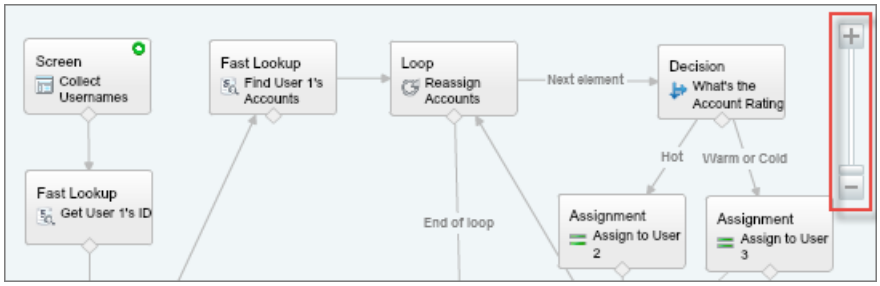
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Zoom

To zoom in and out of your flow, use the + and - buttons on the right side of the canvas.

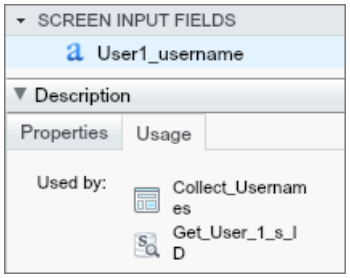


Search in the Explorer tab

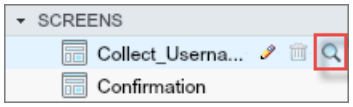
Looking for a specific element or resource? Search for it in the Explorer tab.

- To find an element with a specific name, type in the search box.
- To find all instances of a certain element or resource, click the magnifying glass and select the type.

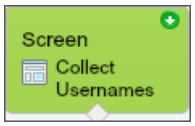
Once you find the right resource in the Explorer tab, see which elements are using the resource. In the Description pane, click the Usage tab.



Once you find the right element in the Explorer, find that element in your canvas. Hover over the element, and click the magnifying glass.



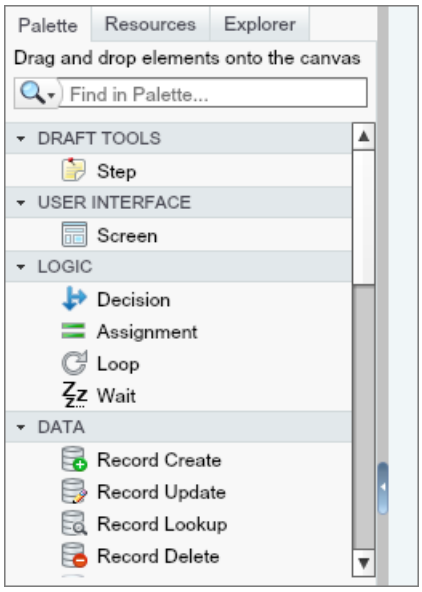
The element is highlighted in green in your canvas.



If the element wasn't in view, the Cloud Flow Designer automatically scrolls to show the element.

Collapse the left side panel

To hide the Palette, Resources, and Explorer tabs from your view, click the left arrow next to the side panel. That way, you get even more space in the canvas.



Considerations for Two-Column Flows (Beta)

If your org has Lightning runtime enabled, you can control whether a flow displays in one column or two columns when you distribute that flow. Before you use this beta feature, though, understand how flow layout currently behaves.

Note: This release contains a beta version of Two-Column Flows, which means it's a high-quality feature with known limitations. Two-Column Flows isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for Two-Column Flows in the IdeaExchange.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Granularity

The layout setting is applied at the flow level. So you can't control the layout at the screen or field level. If you set a flow to use two columns, every screen in that flow displays in two columns.

Order of Fields

You can't manually control which fields go in which columns. If the flow is set to display two columns, the fields alternate in each column. The odd fields (first, third, fifth, and so on) are placed in the left column. The even fields (second, fourth, sixth, and so on) are placed in the right column.

If your users navigate screens with the TAB button, they'll tab through all the fields in the left column and then all the fields in the right column. You can't configure the fields to tab left-to-right.

Responsiveness

The flow layout isn't responsive to the user's screen dimensions. It uses the same layout whether the user's screen is one inch wide or twenty inches wide.

Tip: Don't apply two-column layout to a flow if users will run it from a phone or small tablet.

SEE ALSO:

Limits and Considerations for Visual Workflow

Considerations and Limitations for Flows in Lightning Pages (Beta)

Render Two-Column Screens from a Flow URL (Beta)

Limitations for Multi-Select Choice Fields

Multi-select checkboxes and multi-select picklist fields let flow users select multiple choices in a screen field. Before you start using multi-select choice fields, understand how they work in flows, both when you design the flows and when your users run them.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Configuring a Multi-Select Resource Field

- A multi-select choice field can have only one default value.
- A dynamic record choice resource can be configured to assign field values from a user-selected record to variables in the flow. When a multi-select choice field uses a dynamic record choice, only values from the last record that the user selects are stored in the flow variables. If multiple multi-select choice fields on one screen use the same dynamic record choice, the variable assignments obey the first of those fields.

Using Values from a Multi-Select Resource Field

- At run time, a multi-select field's value is a concatenation of the user-selected choice values, separated by semicolons. If any of the selected choices' values included semi-colons, those semi-colons are removed.
- If you referenced multi-select choice fields in flow conditions, follow these best practices.
  - Configure a stored value for each choice that you use in multi-select choice fields.
  - Don't use the same choice in multiple multi-select choice fields on the same screen.

SEE ALSO:

Flow Screen Element: Choice Fields

Limitations for Flow Formulas

When you create a formula resource or add validation to a screen input field, understand the formula limitations in Visual Workflow.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Flow formulas can't contain more than 3,000 characters.
- A formula returns `null` if:
  - The value that the formula returns doesn't match its data type.
  - The formula contains an unsupported function.

  For example, if your formula resource has a data type of Number, the output must be numeric.
- These functions aren't supported in a flow formula.
  - GETRECORDIDS
  - IMAGE
  - INCLUDE
  - ISCHANGED
  - ISNEW
  - PARENTGROUPVAL
  - PREVGROUPVAL
  - PRIORVALUE
  - REQUIRE SCRIPT
  - VLOOKUP

  For a complete list of operators and functions for building formulas in Salesforce, see Formula Operators and Functions on page 234.
- In a flow, the `CONTAINS` function checks all characters within its parentheses. For cross object field references, `CONTAINS` works like it does in the rest of Salesforce. It checks only the first 250 characters in the reference.

  Here's an example. `varContract` refers to an sObject variable that contains the values of a contract record. This formula expression checks only the first 250 characters.

  ```
  CONTAINS({!varContract.Account.Description}, "description")
  ```

  This formula expression checks all characters in the field.

  ```
  CONTAINS({!varContract.Description}, "description")
  ```
- If a Display Text screen field contains an invalid formula resource, the flow displays an empty string at run time.
- If a formula expression has an error at run time, it resolves to null.
- If a flow contains an invalid formula resource, you can't activate the flow.

SEE ALSO:

Flow Formula Resource

Flow Resources

Limitations for Time-Based Flows

Before you design flows that contain one or more Wait elements, understand the limitations and guidelines.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- After you deactivate a flow or flow version, the associated waiting interviews continue as usual. You can't delete a flow or flow version if it has associated waiting interviews.
- An interview can execute only one event path per Wait element. After one of its events is processed, the remaining events are removed from the queue.
- An organization can process up to 1,000 events per hour. When an event is processed, the interview that it's associated with is resumed and any other events for that interview are removed from the queue. If an organization exceeds this limit, Salesforce processes the remaining events in the next hour.

  For example, an organization has 1,200 events scheduled to be processed between 4:00 PM and 5:00 PM. Salesforce processes 1,000 events between 4:00 PM and 5:00 PM and the additional 200 events between 5:00 PM and 6:00 PM.
- An organization can have up to 30,000 interviews waiting at a given time.
- If the user who started the interview is deactivated when Salesforce tries to execute an event path, the interview fails to resume.

Transactions and Waiting Interviews

A transaction ends as soon as a flow interview begins to wait for an event. When the flow interview resumes, a new transaction begins. Everything after the Wait element is executed as part of a batch transaction that includes other resumed interviews.

Interviews aren't resumed independently. They're grouped into a single batch that starts resuming within one hour after the first interview enters the batch. Any actions that fire as a result of those grouped interviews are also executed in that transaction. This behavior can cause you to exceed your Apex governor limits if the resumed interview executes DML operations or SOQL queries through:

- Flow elements such as Record Create or Fast Lookup
- Apex Plug-in elements
- Apex triggers
- Immediate workflow actions

For details on Apex governor limits, see Limits for Visual Workflow on page 464.

If a Wait element precedes any flow elements that execute DML operations or SOQL queries:

- Ensure that your flows don't perform more DML operations or SOQL queries between Wait elements than the Apex governor limits allow.
- Consider using multiple Wait elements so that the DML operations and SOQL queries are performed in multiple transactions.
- Add fault paths for those elements so that the flow returns to the Wait element if the fault message contains:

  ```
  Too many SOQL queries
  ```

  or

  ```
  Too many DML operations
  ```

If an interview fails after it's resumed:

- Prior interviews in that batch's transaction are successful.
- Operations that the interview executed before it waited are successful.
- If a fault connector handles the failure, operations that the interview executed between when it resumed and when it failed are successful. The operation that caused the interview to fail isn't successful.

- If a fault connector doesn't handle the failure, operations that the interview executed between when it resumed and when it failed are rolled back. The operation that caused the interview to fail isn't successful.
- The remaining interviews in that batch are tried.

Limitations for General Alarms

- Alarms don't support minutes or seconds.
- If an interview is waiting for an event that's set for a time in the past, Salesforce resumes the interview within one hour.

  For example, a flow is configured to email an opportunity owner seven days before the close date. An interview is started for an opportunity with the close date set to today. Salesforce resumes the interview within an hour.

Limitations for Absolute Time Alarms

- Absolute time alarms are evaluated based on the time zone of the user who created the flow.

Limitations for Relative Time Alarms

- Relative time alarms are evaluated based on the organization's time zone.
- Across all your flow versions, your organization can have up to 20,000 defined relative time alarms.
- Alarms can't reference the following:
  - `DATE` or `DATETIME` fields that contain automatically derived functions, such as *`TODAY`* or *`NOW`*.
  - Formula fields that include related-object merge fields.
- If you change a date field that's referenced by an unexecuted relative time alarm in a waiting interview, Salesforce recalculates the events associated with the interview.

  For example, a flow is configured to email an opportunity owner seven days before the opportunity close date and the close date is 2/20/2014. The following things could happen.
  - The close date isn't updated before the interview resumes. Result: Salesforce resumes the interview on 2/13/2014 and sends the email.
  - The close date is updated to 2/10/2014 before the interview resumes. Result: Salesforce reschedules the relative time alarm and the interview resumes on 2/3/2014.
  - The close date is updated to a date in the past. Result: Salesforce recalculates the relative time alarm and resumes the interview shortly after you save the record.
- If a relative time alarm references a null date field when the interview executes the Wait element, Salesforce resumes the interview within an hour.
- If a relative time alarm references a date field that's that has a non-null value when the flow interview executes the Wait element and it's updated to `null` before the alarm is processed, Salesforce resumes the interview within an hour after the date field is updated.
- If a waiting interview has a relative time alarm and the referenced record or object is deleted, the alarm is removed from the queue. If the interview has no other events to wait for, the interview is deleted.
- You can't archive a product or price book that's referenced in a relative or absolute time alarm in a waiting interview.
- Lead Convert Limitations
  - You can't convert a lead that has associated relative time alarms in waiting interviews.
  - If Validation and Triggers from Lead Convert is enabled, existing operations on leads after a Wait element aren't executed during lead conversion.

- If a campaign member based on a lead is converted before a waiting interview that's associated with that record finishes, Salesforce still executes the interview.

SEE ALSO:

Considerations for Designing Flows

Limits and Considerations for Visual Workflow

Operators in Flow Conditions

Flow Wait Element

Considerations for Managing Flows

When managing flows, consider the administration and activation limits.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Activating Flows

When you activate a new version of a flow, the previously activated version (if one exists) is automatically deactivated. Any running flow interview continues to run using the version with which it was initiated.

Deleting Flows

To delete an active flow version, first deactivate it. If a flow has any paused or waiting interviews, it can't be deleted until those interviews are finished or deleted. Flows that have never been activated can be deleted immediately.

Flow Properties

The properties for a given flow's versions automatically match the active version's properties by default. In other words, if you have three versions and you activate version 2, Salesforce updates the properties for versions 1 and 3 to match version 2. However, if you edit the properties for an inactive version, that version's properties are no longer automatically updated to match the active version.

The flow's active (or latest) version determines the flow's type. For example, if a flow's active version contains a screen, its type is Flow. It can't be implemented through a system-based method, like the Process Builder.

IN THIS SECTION:

Considerations for Installed Flows

Keep these considerations in mind when you distribute, upgrade, or remove a flow that you installed from a package.

SEE ALSO:

Manage Your Flows

Limits and Considerations for Visual Workflow

Considerations for Installed Flows

Keep these considerations in mind when you distribute, upgrade, or remove a flow that you installed from a package.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- The Cloud Flow Designer can't open flows that are installed from managed packages.
- If you install a package that contains multiple flow versions in a fresh destination organization, only the latest flow version is deployed.
- If you install a flow from a managed package, error emails for that flow's interviews don't include any details about the individual flow elements. The email is sent to the user who installed the flow.
- If you install a flow from an unmanaged package that has the same name but a different version number as a flow in your organization, the newly installed flow becomes the latest version of the existing flow. However, if the packaged flow has the same name and version number as a flow already in your organization, the package install fails. You can't overwrite a flow.

Status

An active flow in a package is active after it's installed. The previous active version of the flow in the destination organization is deactivated in favor of the newly installed version. Any in-progress flows based on the now-deactivated version continue to run without interruption but reflect the previous version of the flow.

Distributing Installed Flows

- When you create a custom button, link, or Web tab for a flow that's installed from a managed package, include the namespace in the URL. The URL format is `/flow/namespace/flowuniquename`.
- When you embed a flow that's installed from a managed package in a Visualforce page, set the name attribute to this format: `namespace.flowuniquename`.

Upgrading Installed Flows

Upgrading a managed package in your organization installs a new flow version only if there's a newer flow version from the developer. After several upgrades, you can end up with multiple flow versions.

Removing Installed Flows

- You can't delete a flow from an installed package. To remove a packaged flow from your organization, deactivate it and then uninstall the package.
- You can't delete flow components from Managed - Beta package installations in development organizations.
- If you have multiple versions of a flow installed from multiple unmanaged packages, you can't remove only one version by uninstalling its package. Uninstalling a package—managed or unmanaged—that contains a single version of the flow removes the entire flow, including all versions.

SEE ALSO:

Flows in Change Sets and Packages

Considerations for Deploying Flows with Packages

Considerations for Running Flows

When you run or test a flow, keep certain limits and guidelines in mind.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Be careful when testing flows that contain delete elements. Even if the flow is inactive, it triggers the delete operation.
- At run time, time zones for date/time values can differ from what you see in the Cloud Flow Designer. During run time, date/time values reflect the running user's time zone settings in Salesforce.
- Interviews don't perform actions—such as sending emails or creating, editing, or deleting records—until the associated transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element. Record and Fast elements aren't the only ones to create or update records. Post to Chatter, Submit for Approval, and Quick Actions elements do, as well.
- Don't use your browser's Back or Forward buttons to navigate through a flow. Doing so can result in inconsistent data between the flow and Salesforce.
- A single flow can have up to 50 different versions. When you run a flow, you see the active version, but your admin could have a more recent version.
- For flows that interact with the Salesforce database, make sure that your users have permission to create, read, edit, and delete the relevant records and fields. Otherwise, users receive an insufficient privileges error when they try to launch a flow. For example, a flow looks up and updates a case record's status. The flow users must have "Read" and "Edit" permissions on the `Status` field of the Case object.
- When you distribute a flow, don't pass a currency field value from a Salesforce record into a flow Currency variable with a URL parameter. When a currency field is referenced through a merge field (such as `{!Account.AnnualRevenue}`), the value includes the unit of currency's symbol (for example, $). Flow variables of type Currency can accept only numeric values, so the flow fails at run time. Instead, pass the record's ID to a flow Text variable with a URL parameter. Then in the flow, use the ID to look up that record's value for the currency field.

Lightning Runtime Beta Limitations

Note: This release contains a beta version of Flow Lightning Runtime, which means it's a high-quality feature with known limitations. Flow Lightning Runtime isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for the Flow Lightning Runtime in the IdeaExchange.

When Lightning runtime is enabled for your org, flows in Lightning Experience don't load:

- In Web tabs
- In list buttons that are set to display an existing window with or without sidebar

When Lightning runtime is enabled for your org, flows in Salesforce Classic don't load:

- In Web tabs
- In custom buttons or links that are set to display in existing window with or without sidebar

Users can't enter more than 16 total digits, including digits before and after a decimal point.

SEE ALSO:

- Test a Flow
- Flow Interviews
- Limits and Considerations for Visual Workflow
- Flow Runtime Experiences (Beta)

Flow Accessibility

Visual Workflow is 508-compliant with a few exceptions.

- The title of the screen doesn't change when you click Next or Previous, so you might not realize you're on a new page.
- Radio button fields don't have labels. Screen readers can't distinguish between questions.
- Questions without defined prompts can read incorrectly.
- Errors are not noted when reading the fields.

SEE ALSO:

- Limits and Considerations for Visual Workflow

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Create a Flow

Once you understand the process that you want to automate, design a flow in the Cloud Flow Designer for that process.

Tip: Before you start creating your flow, plan it out. It's much easier to automate a business process by using Visual Workflow when you fully understand the details of your business process.

If you're new to the Cloud Flow Designer, we recommend walking through one or more of the sample flow tutorials in the *Cloud Flow Designer Workbook*. They're a great way to learn about the tool and discover how it works.

1. Open the Cloud Flow Designer. From Setup, enter *Flows* in the Quick Find box, then select **Flows**, and then click **New Flow**.
2. Drag the appropriate elements onto the canvas.

   Tip: If you're not sure which element you need for a node, add a Step element as a placeholder until you figure it out. You can always replace the Step later.
3. Connect the elements together so that it's clear what the order of the elements is.
4. Identify which element the flow should start with when it runs.
5. Save any changes that you made to the flow.
6. Test the flow to make sure it's working as you expect it to.
7. Activate the flow so that users can run it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

8. Distribute the flow to the appropriate users.

SEE ALSO:

- Manage Your Flows
- Considerations for Designing Flows
- Flow Accessibility
- Flow Building Blocks

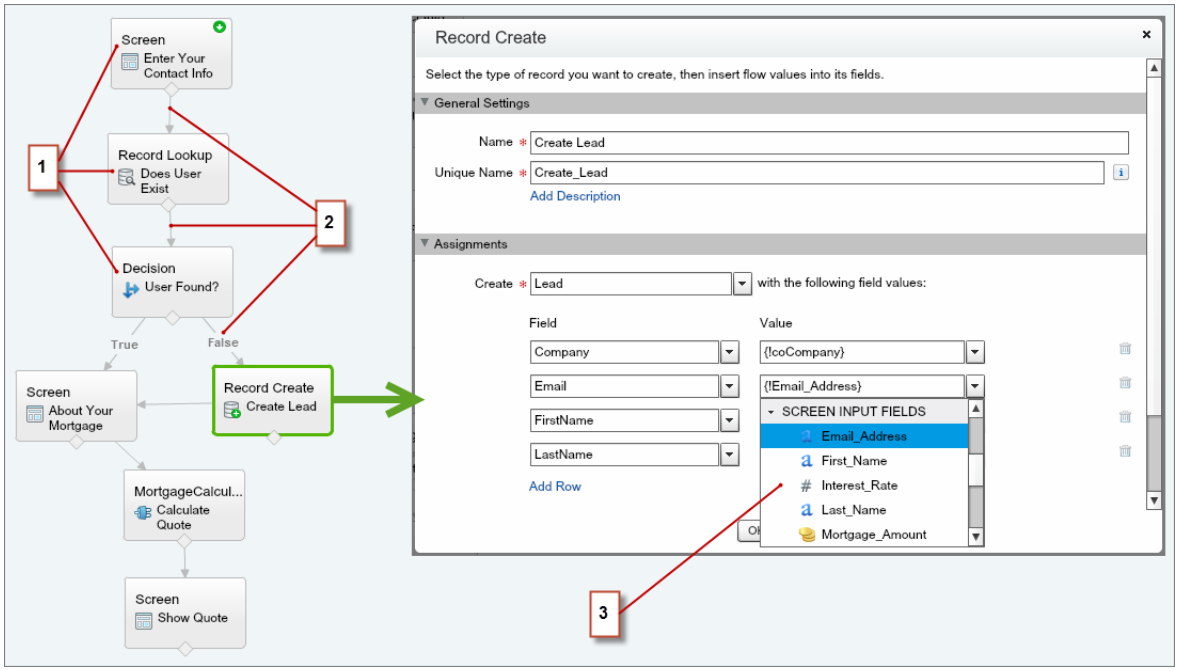
Flow Building Blocks

Use combinations of elements, connectors, and resources to build flows.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



- Each element (1) represents an action that the flow can execute. Examples include reading or writing Salesforce data, displaying information to and collecting data from flow users, executing logic, or manipulating data.
- Each connector (2) defines an available path that the flow can take at run time.

- Each resource (3) represents a value that you can reference throughout the flow.

SEE ALSO:

Flow Elements

Flow Resources

Flow Connectors

Cloud Flow Designer

The Cloud Flow Designer lets you design flows without writing any code.

Watch a Demo: Visual Workflow Cloud Flow Designer Overview

For a collection of useful resources, including videos and sample flows, open the Cloud Flow Designer and click **Get Started**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Requirements for the Cloud Flow Designer

To use the Cloud Flow Designer, you need an up-to-date browser and Adobe® Flash® Player.

Tour the Cloud Flow Designer User Interface

Before you use the Cloud Flow Designer to design flows, understand the tool's main components.

Search Within a Flow

As a flow grows and becomes more complex, it becomes more challenging to find things within it. The Cloud Flow Designer offers tools for quickly finding flow elements and resources.

Search Within the Palette

As you add more flows, actions, and Apex classes to your organization, it becomes more challenging to find a specific item in the Palette. You can, however, search in the Palette to quickly find the right element for your flow.

SEE ALSO:

Flow Elements

Flow Resources

Requirements for the Cloud Flow Designer

To use the Cloud Flow Designer, you need an up-to-date browser and Adobe® Flash® Player.

We recommend:

- Windows® Internet Explorer® versions 8 through 11, Google® Chrome™, or Mozilla® Firefox®. Internet Explorer 6 and 7 are not supported.
- Adobe® Flash® Player version 10.1 and later. The minimum version required to run the Cloud Flow Designer is 10.0.
- A minimum browser resolution of 1024x768.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

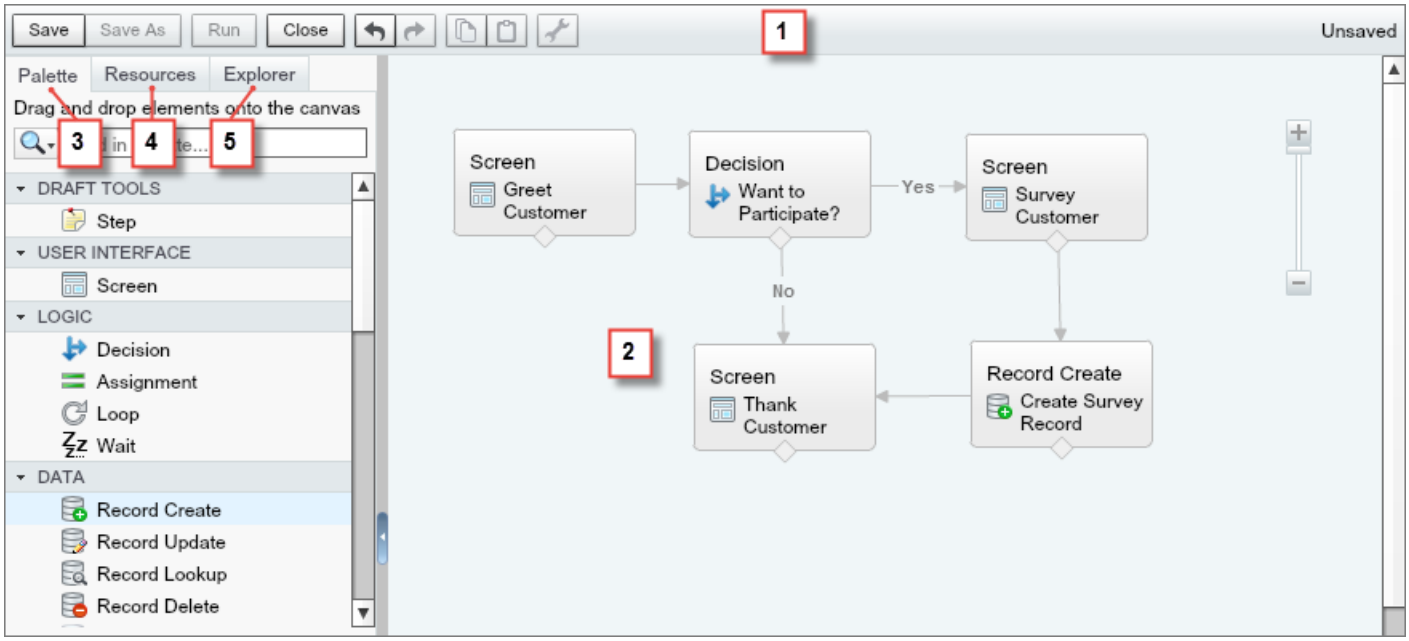
Tour the Cloud Flow Designer User Interface

Before you use the Cloud Flow Designer to design flows, understand the tool's main components.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



Button Bar (1)

Manage your flow as you build it.

- **Run** runs the most recent saved version of the flow that you have open. If the flow contains subflow elements, each subflow runs the active version of its referenced flow. If the referenced flow has no active version, then the subflow element runs the latest version of its referenced flow.
- The status indicator on the right side displays whether:
  - The flow is active or not
  - The latest changes to the flow are saved or not
  - There are any warnings or errors in the last saved version of the flow

    To see a list of the warnings or errors, click the indicator.

Canvas (2)

The canvas is the working area, where you build a flow by adding elements. As you add elements to the canvas and connect them together, you see a visual diagram of your flow.

Palette Tab (3)

Add new elements, like Screens and Record Creates, to your flow from the Palette tab.

Resources Tab (4)
Create resources, like a variable or formula, to use in your flow from the Resources tab.

Explorer Tab (5)
The Explorer tab is a library of all elements and resources that you've added to the flow.

SEE ALSO:

Flow Properties

Manage Flow Elements, Resources, and Connectors

Search Within the Palette

Search Within a Flow

Search Within a Flow

As a flow grows and becomes more complex, it becomes more challenging to find things within it. The Cloud Flow Designer offers tools for quickly finding flow elements and resources.

Open the flow in the Cloud Flow Designer. Then find an element or resource in the flow by using one or more of the following options.

- On the Explorer tab, enter search text.
  The Explorer tab displays only the elements and resources whose properties contain the entered text.
- Filter the Explorer tab contents to one type of element or resource by clicking .
  To remove the filter, click and select **SEARCH ALL**.
- Dim all visible elements on the canvas other than the results by selecting **Highlight Results on Canvas**.
- Zoom in and out as desired using the controls near the top right corner of the canvas area.
- To see the location of an Explorer item on the canvas, complete one of the following procedures.
  If the Explorer item is a canvas-visible element or a screen field:
  1. Hover over the item on the Explorer tab.
  2. Click .

  If the Explorer item is a resource that doesn't appear on the canvas:
  1. Click the item on the Explorer tab.
  2. Click the Usage tab in the Description pane.
  3. Hover over an element listed on the Usage tab.
  4. Click its .

  The canvas shifts to display the element and momentarily highlights it.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:
- "Manage Force.com Flow"

SEE ALSO:

Tour the Cloud Flow Designer User Interface

Search Within the Palette

As you add more flows, actions, and Apex classes to your organization, it becomes more challenging to find a specific item in the Palette. You can, however, search in the Palette to quickly find the right element for your flow.

From the Palette tab, use the following options to find a specific item.

- Next to , enter search text.
  The Palette displays only the items that contain the entered text.
- To filter the Palette tab contents to one type of element, click and select what you want to see.
- To remove the filter, click and select **SEARCH ALL**.

SEE ALSO:

Tour the Cloud Flow Designer User Interface

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Set a Flow's Start Element

Before you can save a flow, indicate which element to execute first.

1. Hover over the starting element in your flow.
2. Click .

SEE ALSO:

Save a Flow

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Save a Flow

After you create a flow in the Cloud Flow Designer, you have some options for saving the flow.

Initial save
When you save a new flow for the first time, a dialog box appears. Enter values for each of the flow's properties. Once you save the flow, the unique name can't be changed.

Quick save
After you've saved a flow once, the **Save** button works as a quick-save, overwriting your previous work. However, the **Save** button doesn't work when editing active flows. To save your changes as a new version or new flow, use **Save As**.

Save As
After you've saved your flow once, this button is enabled with two options:

- **Save as new flow** opens a dialog box where you can input a new name, unique name, and description, then save your changes as an entirely new flow.
- **Save as new version** saves the flow as a new version of the current flow. Use this option if you want to change a flow and keep the old configuration as a backup.

  Each flow can have up to 50 versions. You can't update the unique name when you save a new version.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

When saving a flow or flow version:

- If you have the flow detail page open in one browser tab, then edit a version in another tab, before you run the edited version:
  1. Save the version.
  2. Close the Cloud Flow Designer.
  3. Refresh the flow detail page in the first tab.
- If you've changed the flow properties and for some reason the flow fails to save, the flow properties don't revert to the previous values.

SEE ALSO:

Cloud Flow Designer

Activate or Deactivate a Flow Version

Common Flow Tasks

A handful of tasks are common to multiple flow use cases. For example, you can define conditions in both Decision and Wait elements.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Manage Flow Elements, Resources, and Connectors

Customize your flow by adding, editing, or removing elements, resources, and connectors.

Working with Salesforce Records in a Flow

The real power of a flow is that it can automate updates to your organization's records. In a flow, you can automatically look up values from records, create records, update records, delete records—the whole shebang!

[Validate Users' Inputs with Flow Formulas]
Just like with regular validation rules, you can validate what users enter in flow screens.

[Define Flow Conditions]
Control when a flow takes a specific decision outcome or waits for a specific wait event.

[Define the Path That a Flow Takes]
Identify which elements the flow executes and in what order by connecting the elements on your canvas together.

[Customize What Happens When a Flow Fails]
If your flow contains an element that interacts with the Salesforce database—such as a Record Update or Submit for Approval element, it can fail. Modify the default behavior by adding fault paths to all elements that can fail.

Manage Flow Elements, Resources, and Connectors

Customize your flow by adding, editing, or removing elements, resources, and connectors.

	Add	Edit	Remove
Element	Drag from the Palette tab and drop it on to the canvas.	Double-click, or hover over it and click .	Hover over it and click .
Resource	From the Resources tab, double-click.	From the Explorer tab, double-click or hover over it and click .	From the Explorer tab, hover over it and click .
Connector	Click the node at the bottom of an element on the canvas and drag a line anywhere onto the target element.	n/a	Select it and press the DELETE key.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:
- "Manage Force.com Flow"

SEE ALSO:

[Flow Elements]
[Flow Resources]
[Flow Connectors]

Working with Salesforce Records in a Flow

The real power of a flow is that it can automate updates to your organization's records. In a flow, you can automatically look up values from records, create records, update records, delete records—the whole shebang!

For each of those operations, the Cloud Flow Designer offers at least two elements to choose from. Review the following topics to understand the differences between those elements and decide which one is best for your use case.

Tip: Be familiar with the API names for the objects and fields that you want to work with. The Cloud Flow Designer displays API names instead of labels.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Pull Values from Salesforce Records into a Flow

Before you can use information from your Salesforce records in a flow, pull that information into variables in your flow. Use either a Record Lookup element or a Fast Lookup element. The right element depends on what the rest of your flow is doing.

Alternatively, pass values in from an element that interacts with the Salesforce database—such as the ID of the post created by a Post to Chatter element.

Example: You need to email a given account's owner. To do so, the flow needs to know the email address and name of that user.

To pull values into a flow from records in your organization, use either the **Record Lookup** or **Fast Lookup** element in the Cloud Flow Designer.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How do I choose between flow lookup elements?

The two flow lookup elements are pretty similar. This table summarizes the two main differences between them.

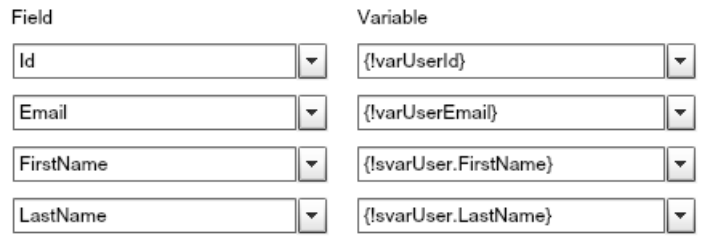
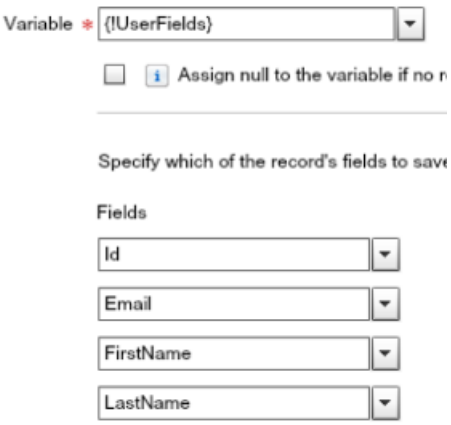
	Can store values in ...	To map field values to flow variables ...	Number of records it looks up
Record Lookup	• Variables • sObject variables	1. Identify each field that you want to store. 2. For each field, identify a flow variable to store that specific value in. Because you directly map each field value to a variable, you get more granularity with this element. However, with more granularity comes more clicking.	Exactly one.
Fast Lookup	• sObject variables • sObject collection variables	1. Identify the flow variable in which you want to store all field values. 2. Identify the fields whose values you want to store in that flow variable.	If an sObject variable: one. If an sObject collection variable: at least one.

Unless you want to map each field to a variable with fewer mouse clicks, it can be hard to choose between the two elements. To choose the right lookup element, figure out what type of variable you need to store the values in.

- To store the values in a single-value non-sObject variable, use the Record Lookup element.
- To store the values in an sObject collection variable, use the Fast Lookup element.
- To store the values in a single-value sObject variable, it's your choice. (Fast Lookup might save you some clicks!)

Tip: It's best practice to use Fast elements whenever possible, so that you save your org's limits. For more information, see Flow Bulkification in Transactions.

Example: Here's how you'd store a user's email and name by using each of the lookup elements.

Record Lookup	
Fast Lookup	

SEE ALSO:

Flow Fast Lookup Element

Flow Record Lookup Element

Working with Salesforce Records in a Flow

Create Salesforce Records from a Flow

To create Salesforce records, use either the Record Create, Quick Action, or Fast Create element. The right element depends on what the rest of your flow is doing.

Example: When the customer's satisfaction score drops below a certain number, automatically create a case.

To create one or more Salesforce records, your flow:

1. Identifies the field values for the new records.
2. Saves those changes to the Salesforce database. (In other words, until the changes are saved to the database, the changes exist only within the flow.)

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How do I choose between flow elements that create records?

The main difference between create elements lies in how many records the element can create and how it knows the field values to apply.

I need to create more than one record at a time.

To create more than one record at a time, use a Fast Create element with an sObject collection variable. It's best practice to use Fast elements whenever possible, so that you stay within your org's limits. For more information, see Flow Bulkification in Transactions.

Record Create and Quick Action elements can create only one record at a time. Fast Create elements can create either one record (if using an sObject variable) or multiple records (if using an sObject collection variable).

I need to create exactly one record.

If you've already populated an sObject variable with the values you want your record to have, use a Fast Create element.

If you want to use a combination of the values from an sObject variable and values from other resources (like single-value variables or screen input fields), use either a Record Create or Quick Action element. Those two elements differ in these ways.

- Which fields are available in the elements
- Whether the element provides any required fields for the object
- Whether the element lets you store the new record's ID

  Storing the ID is useful, for example, if you create an account and then want to create a contact that's associated with that account (which you obviously need the ID for).

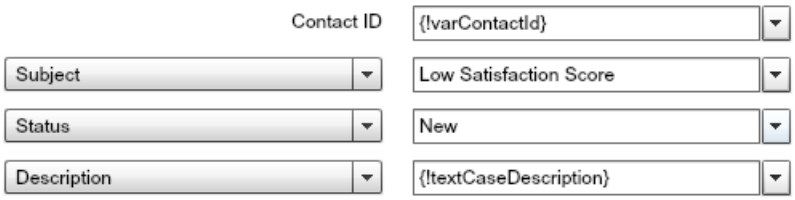
	Field Availability	Required Fields	New Record ID
Record Create	Every field on the object. You manually select the object and every field you want to have a value.	Not indicated	Lets you store the ID of the created record to use later in your flow.
Quick Action (of type Create)	Only fields that are included in the Quick Action layout. If you supplied default values for certain fields when you created the quick action, those values are used when the record is created.	Indicated Requiredness is based on what's marked required in the quick action layout.	Doesn't let you store the created record's ID for use later.

Tip: Use the Quick Actions element when all these statements are true.

1. The action is of type Create.
2. The action's layout includes all the fields that you want to update.
3. You don't need to reference the new record's ID later in the flow.

Otherwise, use the Record Create element.

Example: Here's how you'd create a case when a customer's satisfaction score is too low by using each of the create elements.

Record Create	 You can set any field on the record, but the Record Create element doesn't know which fields are required for this object.
Fast Create	 Assumes {!svarCase} is already populated with the right fields.
Quick Action (of type Create)	 These four fields are the only fields that you can set for this element, because they're the only ones available from the action layout. Contact ID is required by the associated action layout, so it's required in this element.

SEE ALSO:

- Flow Fast Create Element
- Working with Salesforce Records in a Flow
- Flow Quick Action Element
- Flow Record Create Element

Update Salesforce Records from a Flow

To update field values on existing Salesforce records, use either the Record Update, Quick Action, or Fast Update element. The right element depends on what the rest of your flow is doing.

Example: On an opportunity record, when a user clicks the "Won" button, a flow updates the opportunity's stage.

To update fields on one or more existing Salesforce records, your flow:

1. Identifies the records to update.
2. Identifies the new field values for those records.
3. Saves those changes to the Salesforce database. (In other words, until the changes are saved to the database, the changes exist only within the flow.)

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How do I choose between flow elements that update records?

The main difference between the elements lies in these areas: how it knows which records to update, how it knows the new field values to apply, and how many records it can update.

Quick Action elements can update only one record at a time, while Record Update and Fast Update elements can update multiple records.

	To identify records to update	To identify new field values for the records	Number of records it updates
Record Update	In the same element, use filter criteria.	In the same element, map each field that should be updated with a variable or other resource. All resources are supported, so long as the resource's data type matches the selected field's data type.	At least one.
Quick Action	Populate a single-value variable with the ID in another element. Use this ID for the Related Record ID parameter.	In the same element, map each field that should be updated with a variable or other resource. All resources are supported, so long as the resource's data type matches the selected field's data type.	Exactly one.
Fast Update	Populate an sObject variable or sObject collection variable in another element	In another element, such as an Assignment element, update the values in the sObject variable or sObject collection variable.	If an sObject variable: one. If an sObject collection variable: at least one.

If the following statement is true, use a Fast Update element:

- You've already populated an sObject variable or sObject collection variable with the values you want:

  Tip:

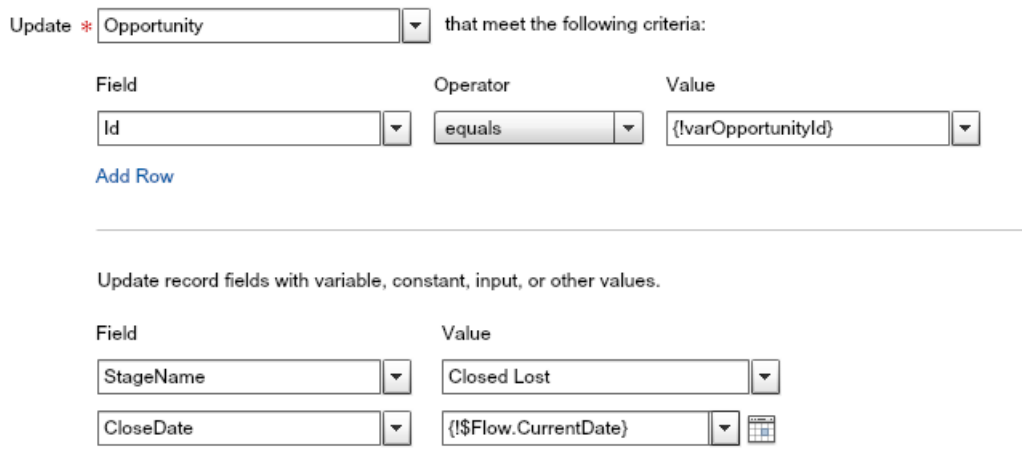
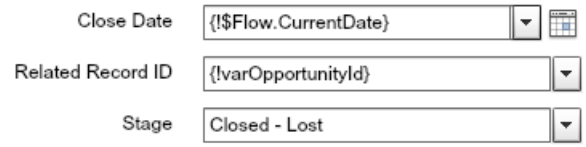
  - You can always update the field values in an sObject variable or sObject collection variable by using an Assignment element.
  - It's best practice to use Fast elements whenever possible, so that you save your org's limits. For more information, see Flow Bulkification in Transactions.

If all the following statements are true, use a Quick Action element:

- You need to update exactly one record
- You've already populated a variable with the record's ID
- The Quick Action's layout includes all the fields you need to update

If any of those statements aren't true, use a Record Update element.

Example: Here's how you'd update an opportunity's stage by using each of the update elements.

Record Update	 You can update any field on the record, but the Record Update element doesn't know which fields are required for this object.
Fast Update	 Assumes {!svarOpportunity} is already populated with the right fields.
Quick Action (of type Update)	 These three fields are required by the associated action layout, so they're required in this element. Related Record ID identifies which opportunity to update.

SEE ALSO:

- Flow Fast Update Element
- Flow Record Update Element
- Flow Quick Action Element
- Working with Salesforce Records in a Flow

Delete Salesforce Records from a Flow

To delete Salesforce records, use either the Record Delete or Fast Delete element. The right element depends on what the rest of your flow is doing.

Example: When a customer accepts a quote, automatically delete the remaining quotes from the opportunity.

To delete one or more records, your flow:

1. Identifies the records that to delete.
2. Saves those changes to the Salesforce database. (In other words, until the changes are saved to the database, the changes exist only within the flow.)

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How do I choose between flow elements that delete records?

The main difference between elements lies in how the element knows which records to delete.

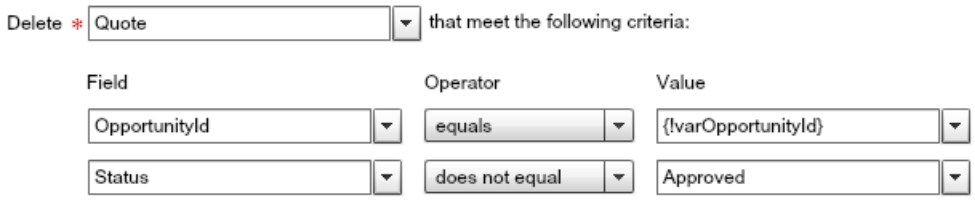
	To identify records to delete
Record Delete	In the same element, use filter criteria.
Fast Delete	In another element, populate an sObject variable or sObject collection variable with the ID of the record to be deleted.

If you've already populated an sObject variable or sObject collection variable with the records you want to delete, use a Fast Delete. (sObject collection variables are supported for record deletion only with a Fast Create element.) It's best practice to use Fast elements whenever possible, so that you save your org's limits. For more information, see Flow Bulkification in Transactions.

If you haven't yet identified which records to delete or you've stored the IDs in non-sObject resources—such as a single-value variable—use a Record Delete element. sObject collection variables aren't supported for this element.

Example: Here's how you'd delete remaining quotes from an opportunity by using each of the delete elements.

Record Delete



The flow finds all quotes that are associated with a specific opportunity and haven't been approved, and then deletes them.

Fast Delete	 Assumes {!svarQuotesUnnecessary} is already populated with the IDs of the quotes to delete. The flow deletes all records whose IDs are included in that variable.

SEE ALSO:

- Flow Fast Delete Element
- Flow Record Delete Element
- Working with Salesforce Records in a Flow

Validate Users' Inputs with Flow Formulas

Just like with regular validation rules, you can validate what users enter in flow screens.

- The formula expression must return a Boolean value (`TRUE` or `FALSE`).
- If the expression evaluates to `TRUE`, the input is valid. If the expression evaluates to `FALSE`, the error message is displayed to the user.
- If the user leaves the field blank and the field isn't required, the flow doesn't validate the field.

When you configure a screen input field:

1. In the Input Validation section, select `Validate`.
2. Define the values allowed for the field by entering a Boolean formula expression.

   Note:

   - The formula expression must return a Boolean value.
   - If the formula expression evaluates to `TRUE`, the input is valid.
   - If the formula expression evaluates to `FALSE`, the error message is displayed to the user.
   - If the user leaves the field blank, and the field is *not* required, the flow doesn't validate.
3. Customize the error message that appears if the user's input fails validation.

   Click to switch between the plain text editor and the rich text editor. Using the rich text editor saves the content as HTML.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Example:

- Validate the format of an email address:

```
REGEX({!Email_Address},"[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}")
```

- Validate the format of a zip code:

```
REGEX({!Zipcode},"\\d{5}(-\\d{4})?")
```

Define Flow Conditions

Control when a flow takes a specific decision outcome or waits for a specific wait event.

Before you begin, create the Decision or Wait element to add conditions to. To add conditions to a wait event, select **Wait for this event only if additional conditions are met**.

1. Set up the conditions.

At run time, the conditions are evaluated in the order you specify.

Column Header	Description
`Resource`	Flow resource whose value you want to evaluate.
`Operator`	The available operators depend on the data type selected for `Resource`. For details, see Operators in Flow Conditions on page 565.
`Value`	The `Variable` and `Value` in the same row must have compatible data types. Options: • Select an existing flow resource, such as a variable, constant, or user input. • Select CREATE NEW to create a flow resource. • Manually enter a literal value or merge field. Note: When you add or subtract a number from a date value, the date adjusts in days, not hours.

2. Identify the logic between the conditions.

Option	Description
`All conditions must be true (AND)`	If one of the conditions is false, the flow evaluates the next outcome's conditions.
`One condition must be true (OR)`	If one of the conditions is true, the flow immediately takes this outcome's path.
`Advanced logic (Combination of ANDs and ORs)`	Custom logic. When you select this option, provide the customized `Logic` by entering a text string. Use: • Numbers to refer to each condition • AND or OR to identify whether all or just one of the conditions must true • Parentheses to group parts of the string together Tip: If you enter *AND*, it's the same as if you selected **All conditions must be true (AND)**. If you enter *OR*, it's the same as if you selected **One condition must be true (OR)**. If you enter any other logic, make sure that you include a number for each condition.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Option	Description
	For example, for `1 AND (2 OR 3)`, the flow evaluates whether the first condition is true and either the second or third condition is true.

SEE ALSO:

- What Are Waiting Conditions?
- Flow Wait Element
- Flow Decision Element

Send Email from a Flow

To send email from your flow, either call an email alert workflow action or create the email in the flow.

Email Alert element
Sends an email by using a workflow email alert to specify the email template and recipients. The flow provides only the record ID.

Send Email element
Sends an email by manually specifying the subject, body, and recipients in the flow.

SEE ALSO:

- Flow Elements

Invoke Apex Code from a Flow

The Cloud Flow Designer comes with a lot of functionality, but sometimes your flow needs to do more than the default elements allow. In that case, call an Apex class from your flow by using one of two flow elements: Apex Plug-in and Call Apex.

Developers have two options when they're trying to make an Apex class available for a flow.

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

While the `Process.Plugin` interface supports customizing how the class appears in the palette, the `@InvocableMethod` annotation provides more functionality. The following table describes the features supported by each option.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

	`Process.Plugin` **Interface**	`@InvocableMethod` **Annotation**
Apex data type support	Doesn't support: • Blob • Collection • sObject • Time	Doesn't support: • Generic Object • Generic sObject • Sets • Maps • Enums

	`Process.Plugin` Interface	`@InvocableMethod` Annotation
		The Cloud Flow Designer doesn't support mapping an Apex method's input or output parameters to an sObject collection variable.
Bulk operations	Not supported	Supported
Element name in the Cloud Flow Designer	Class name or the value of the `name` property.	Class name
Reusability	Classes with this interface implemented are available in flows	Classes with this annotation implemented are available in: • Flows • Processes • Rest API
Section in the Cloud Flow Designer	Apex Plug-in or the value of the `tag` property.	Apex
More Details in the *Force.com Apex Code Developer's Guide*	Passing Data to a Flow Using the `Process.Plugin` Interface	`InvocableMethod` Annotation and `InvocableVariable` Annotation

Example: To illustrate the difference between these two implementation methods, here are two classes that do the same thing: get an account name from a flow and return that account's ID.

This class implements the `@InvocableMethod` annotation.

```
global class lookUpAccountAnnotation {
   @InvocableMethod
   public static List<String> getAccountIds(List<String> names) {
      List<Id> accountIds = new List<Id>();
      List<Account> accounts = [SELECT Id FROM Account WHERE Name in :names];
      for (Account account : accounts) {
         accountIds.add(account.Id);
      }
      return accountIds;
   }
}
```

This class implements the `Process.Plugin` interface.

```
global class lookUpAccountPlugin implements Process.Plugin {

   global Process.PluginResult invoke(Process.PluginRequest request) {
      String name = (String) request.inputParameters.get('name');
      Account account = [SELECT Id FROM Account WHERE Name = :name LIMIT 1][0];

      Map<String,Object> result = new Map<String,Object>();
```

```
        result.put('accountId', account.Id);
        return new Process.PluginResult(result);
    }

    global Process.PluginDescribeResult describe() {
        Process.PluginDescribeResult result = new Process.PluginDescribeResult();
        result.Name = 'Look Up Account ID By Name';
        result.Tag = 'Account Classes';
        result.inputParameters = new
            List<Process.PluginDescribeResult.InputParameter>{
                new Process.PluginDescribeResult.InputParameter('name',
                Process.PluginDescribeResult.ParameterType.STRING, true)
            };
        result.outputParameters = new
            List<Process.PluginDescribeResult.OutputParameter>{
                new Process.PluginDescribeResult.OutputParameter('accountId',
                Process.PluginDescribeResult.ParameterType.STRING)
                    };
        return result;
    }
}
```

Notice that `lookupAccountAnnotation` is less than half the length (11 lines) of `lookupAccountPlugin` (28 lines). In addition, because the annotation supports bulk operations, `lookupAccountAnnotation` performs one query per batch of interviews. `lookupAccountPlugin` performs one query per interview.

SEE ALSO:

Flow Elements

View Inputs and Outputs of Other Referenced Flow Versions

While configuring a subflow element, view the variables of a specified version of the referenced flow. Doing so lets you configure draft master and referenced flows at the same time.

From a subflow element, you can assign values to only the referenced flow's variables that allow input access. Similarly, you can assign values from only the referenced flow's variables that allow output access. The `Input/Output Type` of the variable determines this access. To change the variable's `Input/Output Type`, open the referenced flow to edit the variable.

By default, this drop-down list contains the variables of the currently active version of the referenced flow. If the referenced flow has no active version, the drop-down list contains the variables of the *latest* version of the referenced flow.

To populate the drop-down lists with the variables of another version of the referenced flow, complete the following steps. Do the same to view the descriptions of the referenced flow's variables.

1. On the subflow overlay, expand the `Input/Output Variable Assignments` section.
2. Click **View input/output of other versions**.
3. Use one or more of the following options.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Option	Description
Select a `Version` number in the left pane.	The Inputs and Outputs tabs display the variables in the selected version of the referenced flow.
Select the **Inputs** tab or the **Outputs** tab.	The tab displays: • The variables available for input or output assignment in the selected `Version` of the referenced flow. • The data type of each variable. • The description, if any, of each variable.
Click **OK**.	The subflow overlay's drop-down lists for selecting the referenced flow's variables are populated with the variables of the selected `Version` of the referenced flow.

When you configure subflow input and output assignments, you can specify variables from any version of the referenced flow. This way, you can develop both the master flow and referenced flow in parallel, while keeping another version of the referenced flow active for its users. When you *save* the master flow, however, the Cloud Flow Designer validates against the currently active version of the referenced flow. If that flow doesn't have an active version, the latest version is validated. If you see validation messages about variables that couldn't be found or that were configured differently in the referenced flow, you can still save the flow. Nevertheless, resolve all validation errors before you *activate* the master flow.

SEE ALSO:

Flow Subflow Element

Add Values to a Collection Variable

After you create a collection variable, populate it with values to reference throughout your flow. You can't use a Record Lookup or Fast Lookup element to populate a collection variable, but there are some workarounds.

To use values from outside the flow, set the collection variable's `Input/Output Type` to "Input" and then use URL parameters, Visualforce controllers, or subflow inputs. When the values are coming from outside the flow, the values can be set only at the start of the flow interview.

To add values that are stored in...	Do this...	For more information
A screen field	Add the field's entered or stored value to a collection variable by using an Assignment element	• Choice fields • Input fields • Output fields • Assignments
A variable	Add the variable's stored value to a collection variable by using an Assignment element	• Variables • Assignments

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

To add values that are stored in...	Do this...	For more information
An sObject variable	Add one of the sObject variable's stored field values to a collection variable by using an Assignment element	• sObject variables • Assignments
An sObject collection variable	Loop through the sObject collection variable. Within the loop, add one of the loop variable's stored field values to a collection variable by using an Assignment element	• sObject collection variables • Loops • Assignments

SEE ALSO:

Flow Collection Variable Resource

Sample Flow That Populates a Collection Variable

Define the Path That a Flow Takes

Identify which elements the flow executes and in what order by connecting the elements on your canvas together.

1. On the canvas, find the node at the bottom of the source element.
2. Drag the node onto the target element.
3. If prompted, select which outcome to assign to the path.

SEE ALSO:

Remove Connectors from a Flow

Flow Connectors

Customize What Happens When a Flow Fails

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Remove Connectors from a Flow

You can't modify a connector's target or source elements, so to change a path, delete the connector and then add a new one.

If you delete a connector for a specific outcome, the outcome isn't deleted from the source element. However, if you delete an outcome from a decision element, the outcome's connector is also deleted.

1. In your flow, select the connector to delete.

   When you select a connector, its color changes from gray to green. If you're having trouble selecting a connector, click and drag an area on the canvas that includes the connector.

2. Press DELETE.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

SEE ALSO:

- Define the Path That a Flow Takes
- Flow Connectors

Customize What Happens When a Flow Fails

If your flow contains an element that interacts with the Salesforce database—such as a Record Update or Submit for Approval element, it can fail. Modify the default behavior by adding fault paths to all elements that can fail.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

What Happens When a Flow Fails?

When you're deciding whether to customize the error handling in your flow, consider how a failed flow behaves by default.

Configure Every Fault Path to Send You an Email (Best Practice)

As a best practice, we recommend configuring the fault connectors in your flow so that you always receive an email when a flow fails. In the email, include the current values of all your flow's resources. The resource values can give you insight into why the flow failed.

Customize the Error Message for Running Flow Users (Best Practice)

As a best practice, we recommend displaying a better message to your user than "An unhandled fault has occurred in this flow". Do this only if the distribution method you're using supports flows that contain screens. In other words, don't do it if your flow is distributed through a process.

Other Examples of Error Handling in Flows

Examples of using fault connectors to handle flow errors include requesting corrections from the user and bypassing the error.

SEE ALSO:

- Flow Connectors
- Flow Elements
- Define the Path That a Flow Takes

What Happens When a Flow Fails?

When you're deciding whether to customize the error handling in your flow, consider how a failed flow behaves by default.

Here's what happens by default.

- This error message displays to the running user—the user who was running the flow.

```
An unhandled fault has occurred in this flow
An unhandled fault has occurred while processing
the flow. Please contact your system administrator
for more information.
```

- The running user can't proceed with the flow or return to a previous part of the flow.
- The admin who created the flow receives a fault email. The email details the element that failed, the error message from that element, and which elements were executed during the failed interview. Here's an example error message that can appear in a fault email.

```
An error occurred at element Fast_Delete_1.
DELETE --- There is nothing in Salesforce matching your
delete criteria.
```

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Customize What Happens When a Flow Fails

Configure Every Fault Path to Send You an Email (Best Practice)

As a best practice, we recommend configuring the fault connectors in your flow so that you always receive an email when a flow fails. In the email, include the current values of all your flow's resources. The resource values can give you insight into why the flow failed.

1. Create a text template that includes the values of all the flow resources.

   Doing so lets you see the exact values of flow variables when the interview failed. Also, if the flow contains screens, you see exactly what the user entered and selected.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

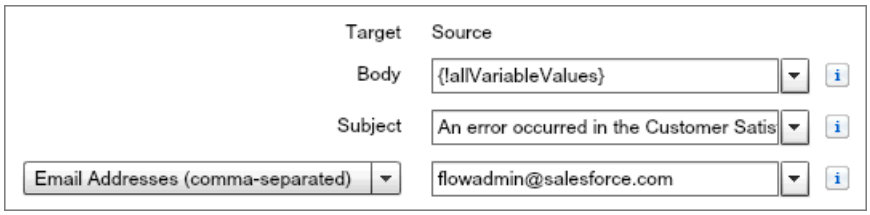
- "Manage Force.com Flow"

Here's an example text template for the Customer Satisfaction Survey flow in the Cloud Flow Designer Workbook.

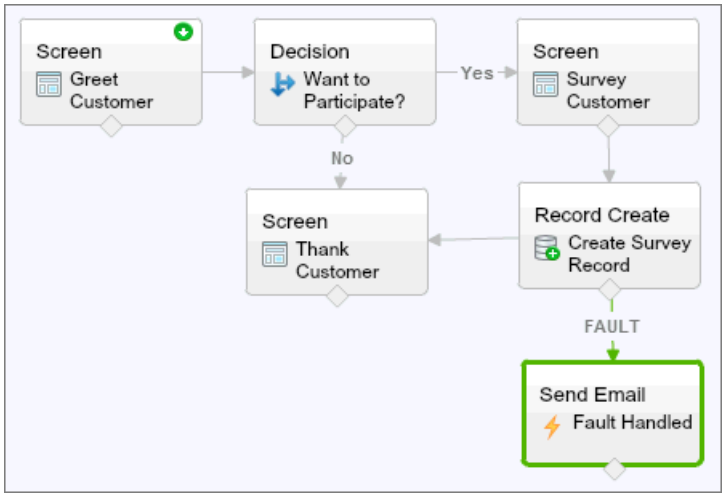
```
Error: {!$Flow.FaultMessage}

RESOURCE VALUES
Customer Response: {!Customer_Response}
Value of Decision's Yes outcome: {!Yes}
Company: {!Company_Name}
Satisfaction Choice Field: {!Satisfaction}
Service Choice Field: {!Service}
Other Comments:
{!OtherComments}
```

2. Configure a Send Email element. Use the text template as the body and your email address as the recipient.
   In this example, `Body` is set to the text template we created: `{!allVariableValues}`.



3. From each element that can fail, draw a fault connecter to the Send Email element.
   In this example, Record Create is the only element that supports fault connectors.



SEE ALSO:

Flow Text Template Resource

Flow Send Email Element

Customize What Happens When a Flow Fails

Customize the Error Message for Running Flow Users (Best Practice)

As a best practice, we recommend displaying a better message to your user than "An unhandled fault has occurred in this flow". Do this only if the distribution method you're using supports flows that contain screens. In other words, don't do it if your flow is distributed through a process.

1. Create a text template that contains a friendlier error message.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

```
<FONT FACE="Arial" STYLE="font-size:14px">
<B>Something went wrong with this flow.</B>
</FONT>
<P>Your admin has received an email about this error.</P>
```

2. Add a Screen element. In a Display Text field, reference the text template.
3. For every element that can fail, draw a fault connector to the Screen element.
   In this example, Record Create is the only element that supports fault connectors. After the flow displays the better error message to the user, it sends an email to the admin with debugging information.

SEE ALSO:

- Flow Screen Element: Display Text Fields
- Flow Text Template Resource
- Customize What Happens When a Flow Fails

Other Examples of Error Handling in Flows

Examples of using fault connectors to handle flow errors include requesting corrections from the user and bypassing the error.

Request Corrections from Users

Draw a fault connector to a Screen element, where users can verify the values that they entered, make corrections, and proceed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Display the Error Message

If the flow is used only internally, such as at a call center, use the fault path to display the error message to the running user. In the same Screen element, ask the user to report the error to the IT department. To do so, draw the fault connector to a Screen element with this Display Text field.

```
Sorry, but you can't read or update records at this time.
Please open a case with IT and include this error message:
{!$Flow.FaultMessage}
```

Create a Case

When an error occurs, automatically create a case that includes the error message and assign it to your IT department. Assign the created case's ID to a Text variable (`{!caseId}`, for example). Then, in a Screen, display this message to the running user.

```
Sorry, but you can't read or update records at this time.
We filed a case for you.
```

Ignore Errors

To bypass errors for a given element in your flow, draw the fault connector to the same element as the normal connector.

SEE ALSO:

Customize What Happens When a Flow Fails

Flow Reference

Bookmark this page for quick access to information about flow elements, resources, events, and more.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Flow Elements

Each *element* represents an action that the flow can execute. Examples of such actions include reading or writing Salesforce data, displaying information and collecting data from flow users, executing business logic, or manipulating data.

Flow Resources

Each *resource* represents a value that you can reference throughout the flow.

Cross-Object Field References in Flows

When building a flow, you can reference fields for records that are related to the values that are stored in an sObject variable. To do so, manually enter the references.

Flow Connectors

Connectors determine the available paths that a flow can take at run time. In the Cloud Flow Designer canvas, a connector looks like an arrow that points from one element to another.

Flow Operators

Operators behave differently, depending on what you're configuring. In Assignment elements, operators let you change resource values. In flow conditions and record filters, operators let you evaluate information and narrow the scope of a flow operation.

Flow Event Types

`Event Type` drives the fields that you use to define an event in a flow Wait element. The available event types are both alarms, which consist of a date/time value—the base time—and an optional offset from that time.

Flow Types

A flow or flow version's type determines which elements and resources are supported, as well as the ways that the flow can be distributed.

Flow Properties

A flow's properties consist of its name, description, interview label, and type. These properties drive the field values that appear on a flow or flow version's detail page. The properties of a flow and its flow versions are separate.

Flow Elements

Each *element* represents an action that the flow can execute. Examples of such actions include reading or writing Salesforce data, displaying information and collecting data from flow users, executing business logic, or manipulating data.

In the Cloud Flow Designer, the canvas and Explorer tab display the elements that exist in the flow. The Palette tab displays the available element types that you can add to the flow by dragging them onto the canvas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

General Settings for Flow Elements

Every flow element has three settings in common: name, unique name, and description.

Flow Apex Plug-In Element

Calls an Apex class that implements the `Process.Plugin` interface. If you used the Tag property in the `PluginDescribeResult` class, the Apex class appears under a customized section. Otherwise, it appears under the Apex Plug-ins section.

Flow Assignment Element

Sets or changes values in variables, collection variables, sObject variables, and sObject collection variables.

Flow Call Apex Element

Calls an Apex class's invocable method.

Flow Decision Element

Evaluates a set of conditions and routes users through the flow based on the outcomes of those conditions. This element performs the equivalent of an if-then statement.

Flow Email Alert Element

Sends an email by using a workflow email alert to specify the email template and recipients. The flow provides only the record ID.

Flow Fast Create Element

Creates Salesforce records using the field values from an sObject collection variable. Or creates one Salesforce record using the field values from an sObject variable.

Flow Fast Delete Element

Deletes Salesforce records using the ID values that are stored in an sObject collection variable. Or deletes one Salesforce record using the ID value that's stored in an sObject variable.

Flow Fast Lookup Element

Finds Salesforce records to assign their field values to an sObject collection variable. Or finds one Salesforce record to assign its field values to an sObject variable.

Flow Fast Update Element

Updates Salesforce records using the field values from an sObject collection variable. Or updates one Salesforce record using the field values from an sObject variable. If a record's ID is included in the variable, its field values are updated to match the other values that are stored in the variable.

Flow Loop Element

Iterates through a collection one item at a time, and executes actions on each item's field values—using other elements within the loop.

Flow Record Create Element

Creates one Salesforce record by using individual field values that you specify.

Flow Record Delete Element

Deletes all Salesforce records that meet specified criteria.

Flow Record Lookup Element

Finds the first Salesforce record that meets specified criteria. Then assigns the record's field values to individual flow variables or individual fields on sObject variables.

Flow Record Update Element

Finds all Salesforce records that meet specified criteria and updates them with individual field values that you specify.

Flow Quick Action Element

Calls an object-specific or global quick action that's already been configured in your organization. Only "Create," "Update," and "Log a Call" actions are supported.

Flow Post to Chatter Element

Posts a message to a specified feed, such as to a Chatter group or a case record. The message can contain mentions and topics, but only text posts are supported.

Flow Screen Element

Displays a screen to the user who is running the flow, which lets you display information to the user or collect information from the user.

Flow Send Email Element

Sends an email by manually specifying the subject, body, and recipients in the flow.

Flow Step Element

Acts as a placeholder when you're not sure which element you need.

Flow Submit for Approval Element

Submits one Salesforce record for approval.

Flow Subflow Element

Calls another flow in your organization. Use this element to reference modular flows and simplify the overall architecture of your flow.

Flow Wait Element

Waits for one or more defined events to occur, which lets you automate processes that require a waiting period.

SEE ALSO:

Flow Resources

Cloud Flow Designer

General Settings for Flow Elements

Every flow element has three settings in common: name, unique name, and description.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Name`	Helps you identify the element on the canvas.
`Unique Name`	Automatically populated if empty when you fill out the `Name` field and press TAB. The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Appears after you click **Add Description**.

Flow Apex Plug-In Element

Calls an Apex class that implements the `Process.Plugin` interface. If you used the Tag property in the `PluginDescribeResult` class, the Apex class appears under a customized section. Otherwise, it appears under the Apex Plug-ins section.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tip:

- Apex classes appear in the palette as Apex plug-ins only if the `Process.Plugin` interface has been implemented.
- If you have many plug-ins in your organization, ask your developer to use the `tag` property. The class appears under a special section header in the palette. Otherwise, the class appears with all the other Apex plug-ins.
- If your developer hasn't already implemented the `Process.Plugin` interface on the desired class, we recommend using the `@InvocableMethod` annotation instead. Unlike the `Process.Plugin` interface, the `@InvocableMethod` annotation supports sObject, Collection, Blob, and Time data types and bulkification. It's also much easier to implement. To see a complete comparison between the interface and the annotation, see Invoke Apex Code from a Flow on page 500.

Inputs

Pass information from the flow to the invoked Apex method. The method determines the available input parameters and their data types.

Outputs

Pass information from the invoked Apex method to the flow. The method determines the available output parameters and their data types.

The flow assigns the values to the specified variables when the code is executed.

Usage

Note: If the Apex class creates, updates, or deletes a record, the action isn't performed until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Invoke Apex Code from a Flow
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows
- *Apex Developer Guide*: Process Namespace

Flow Assignment Element

Sets or changes values in variables, collection variables, sObject variables, and sObject collection variables.

In each row, specify the `Variable` whose value you want to change and the new `Value` for that variable. At run time, the variable assignments occur in the order you specify.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Column Header	Description
`Variable`	Variable whose value you want to change.
`Operator`	The available operators depend on the data type selected for the `Variable`.
`Value`	The `Variable` and `Value` in the same row must have compatible data types. Options: • Select an existing flow resource, such as a variable, constant, or user input. • Select CREATE NEW to create a flow resource. • Manually enter a literal value or merge field.

Example: Change the value of a customer's credit score based on how the customer answered questions in the flow.

SEE ALSO:

- Flow Elements
- Operators in Flow Assignment Elements
- Define the Path That a Flow Takes
- Flow Resources
- Cross-Object Field References in Flows

Flow Call Apex Element

Calls an Apex class's invocable method.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Important: To use this element to call an Apex class from a flow, ask your developer to annotate one of the class's methods with `@InvocableMethod`. For details, see "`InvocableMethod` Annotation" in the *Force.com Apex Code Developer's Guide*.

Inputs

Pass information from the flow to the invoked Apex method. The method determines the available input parameters and their data types.

Outputs

Pass information from the invoked Apex method to the flow. The method determines the available output parameters and their data types.

The flow assigns the values to the specified variables when the method is executed.

Usage

Note: If the invoked method creates, updates, or deletes a record, that action isn't performed until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Invoke Apex Code from a Flow
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Decision Element

Evaluates a set of conditions and routes users through the flow based on the outcomes of those conditions. This element performs the equivalent of an if-then statement.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Outcomes

Create the outcomes for the decision. To rename the path that the flow takes when none of the other outcome conditions are met, click **[Default Outcome]**. Set up the conditions.

Field	Description
`Name`	Identifies the connector for this outcome on the canvas.
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Conditions	Determines whether the flow takes this outcome's path.

Example: Using a Decision element, determine whether:

- To give customers a return shipping address (because an item is definitely faulty) or instructions on how to resolve the problem
- To offer a customer a loan or not (based on results of a credit scoring formula)

Tip: Configure your flow so that it does different things based on which option a user selected for a screen's drop-down list. To do so, add a decision after the screen to create the branches of the flow based on the choices available in that drop-down list. Then you can represent each choice in your decision and connect it to a branch of your flow.

SEE ALSO:

Flow Elements
Define Flow Conditions
Operators in Flow Conditions
Define the Path That a Flow Takes
Cross-Object Field References in Flows

Flow Email Alert Element

Sends an email by using a workflow email alert to specify the email template and recipients. The flow provides only the record ID.

Before you begin:

- Make sure that the email alert you want to call from your flow exists. If not, create the email alert on page 616.
- Understand the daily limits for emails sent from email alerts.
- Store the ID for the record that you want to reference in this email, such as by using a Fast Lookup element. If the email alert has any merge fields, the referenced record is the starting point for those fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The unique name for each email alert is prefixed with its object. The object type of the referenced record must match the object type of the email alert. For example, if you have an email alert with unique name "Owner_Changed" for accounts, that email alert appears in the Palette as `Account.Owner_Changed`. Because the email alert is associated with the Account object, it can reference only an account record.

Field	Description
`Record ID`	Select a variable that contains the ID for the record that you want the email to reference. If the email alert uses any merge fields, this record is the starting point for those merge fields. This field accepts **single-value variables of any type**. The value is treated as text.

Usage

Note: At run time, the email isn't sent until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Send Email from a Flow
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Fast Create Element

Creates Salesforce records using the field values from an sObject collection variable. Or creates one Salesforce record using the field values from an sObject variable.

To create a single record with field values from regular variables and other flow resources, such as constants, formulas, and screen fields, use Record Create.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Variable`	The sObject variable or collection that you want to use to create the record or multiple records. The object types must match, and each ID field must not have a value. This field accepts any **sObject variable or sObject collection variable**.

Example: Take a collection of new cases and use a Fast Create element to create records for each case in the collection. Make sure that your flow populates the sObject variable or collection with all required field values before executing the Fast Create element.

Usage

If you used an sObject variable to create a single record, the sObject variable's ID field is updated with the new record's ID value. If you used an sObject collection to create multiple records, the ID field of each collection item is updated with its matching new record ID value.

Note: At run time, records aren't created until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Create Salesforce Records from a Flow
- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Flow sObject Variable Resource
- Flow sObject Collection Variable Resource

Flow Fast Delete Element

Deletes Salesforce records using the ID values that are stored in an sObject collection variable. Or deletes one Salesforce record using the ID value that's stored in an sObject variable.

Tip: Make sure that the sObject variable or collection is populated with ID values before using the Fast Delete element.

Field	Description
`Variable`	Identifies the sObject variable or collection that you want to use to delete records. The variable must include the IDs of the records that you want to delete. This field accepts any **sObject variable or sObject collection variable**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Usage

Warning:

- Be careful when testing flows that contain delete elements. Even if the flow is inactive, it triggers the delete operation.
- To prevent deleting records by mistake, be as specific in your filter criteria as possible.
- Records are deleted from your organization the moment the flow executes the delete element.
- Deleted records are sent to the Recycle Bin and remain there for 15 days before they are permanently deleted.
- Flows can delete records that are pending approval.

To delete one or more records that meet filter criteria specified by regular variables and other flow resources, such as constants, formulas, and screen fields, use Record Delete.

Note: At run time, the records aren't deleted until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Delete Salesforce Records from a Flow
- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Flow sObject Variable Resource
- Flow sObject Collection Variable Resource

Flow Fast Lookup Element

Finds Salesforce records to assign their field values to an sObject collection variable. Or finds one Salesforce record to assign its field values to an sObject variable.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Look up`	Identifies the object whose records you want to look up.
Filter criteria	Narrows down the scope of records that the flow looks up. Specify the filter criteria for selecting the record from the database. Tip: Make sure that your filter criteria sufficiently narrows the search. If you use an sObject variable to store the results, the Fast Lookup element returns only the first record from the filtered results. `Value` must be compatible with the selected field's data type.
`Sort results by:`	Sorts the filtered results before storing records in the variable. If selected, also select the field that you want to sort the results by and the sort order. Only sortable fields are available.
`Variable`	The variable to contain the returned field values. This field accepts any **sObject variable or sObject collection variable**. • To contain the field values for the first returned record, use an sObject variable. • To contain the field values for all returned records, use an sObject collection variable.
`Assign null to the variable if no records are found.`	Sets the variable to null if no records meet the filter criteria. By default, the variable's values are left unchanged.
`Specify which of the record's fields to save in the variable.`	Identifies which fields on the records that meet the filter criteria to store in the variable. Values for unselected fields are set to null in the variable.

Example:

- Look up a product's name and description by using the bar code on its product tag.
- Look up all customers who live in a particular city.
- Look up customer transactions on a particular day.

Considerations for Defining Filter Criteria

- When you define multiple filters, the filter logic usually defaults to AND. However, if multiple filters have the same field selected and use the equals operator, the filters are combined with OR.

  For example, your filters check whether a case's Type equals Problem (1), Type equals Feature Request (2), and Escalated equals true (3). At runtime, those filters are combined to `(1 OR 2) AND 3`.

- The available filter operators depend on the data type of the selected fields. For details, see Operators in Flow Record Filters.

Usage

To get a single record and store specified field values in regular variables and sObject variables, use Record Lookup.

SEE ALSO:

- Pull Values from Salesforce Records into a Flow
- Flow Elements
- Operators in Flow Record Filters
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Flow sObject Variable Resource
- Flow sObject Collection Variable Resource
- Cross-Object Field References in Flows

Flow Fast Update Element

Updates Salesforce records using the field values from an sObject collection variable. Or updates one Salesforce record using the field values from an sObject variable. If a record's ID is included in the variable, its field values are updated to match the other values that are stored in the variable.

Field	Description
`Variable`	Identifies the sObject variable or collection that you want to use to update records. This field accepts any **sObject variable or sObject collection variable**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: You're designing flows for a call center. To automatically update Salesforce with data collected from callers, such as new addresses or product preferences, use a Fast Update element. Have your flow populate the sObject variable or collection before using the Fast Update element. Then make sure that the sObject variable or sObject values within the collection contain the ID for the records that are being updated.

Usage

To update one or more records with field values from regular variables and other flow resources, such as constants, formulas, and screen fields, use Record Update.

Note: At run time, records aren't updated until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Update Salesforce Records from a Flow
- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Flow sObject Variable Resource
- Flow sObject Collection Variable Resource

Flow Loop Element

Iterates through a collection one item at a time, and executes actions on each item's field values—using other elements within the loop.

A *collection* is a list of items that contain, for example, field values from Salesforce records. A loop uses an sObject variable, referred to as the *loop variable*, to contain the values for the current item in the collection. Once the loop finishes examining an item, it copies the field values for the next item into the loop variable. Then the loop examines those values. The loop variable must have the same object type as the collection. For example, if your collection contains field values from accounts, your loop variable must also be of type Account.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Loop through`	The collection that you want to loop through. This field accepts any **sObject variable or sObject collection variable**.
Order	`Ascending` begins at the start of the collection and moves to the end, while `Descending` begins at the end and moves to the start.
`Loop Variable`	The variable that the flow uses to contain the current item's values during a loop iteration. • If `Loop through` is set to a non-sObject collection variable, this field accepts a single-value variable with the same data type. • If `Loop through` is set to an sObject collection variable, this field accepts an sObject variable with the same object type.

Usage

After you add a Loop element and the elements that you want the loop to include, from the Loop element:

- Determine which element to execute first when a new item's values are copied into the loop variable by adding a "Next element" connector.

- Determine which flow element to execute after the loop has processed all the items in the collection by adding an "End of loop" connector.

SEE ALSO:

Sample Flow That Loops Through a Collection
Define the Path That a Flow Takes
Flow Elements
Flow sObject Collection Variable Resource

Flow Record Create Element

Creates one Salesforce record by using individual field values that you specify.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Create`	The object for which you want to create a record
Field values	Identifies the field values for the new record. Each value must be compatible with the selected field's data type. Important: Ensure that all required fields are populated with values; otherwise the flow fails at run time. If you don't know which fields are required, check the object definition.
`Variable`	Assigns the ID of the new record to a variable so you can reference it later in the flow. This field accepts only **single-value variables of type Text**.

Example: A user enters a name and address into the flow. Verify that a matching user exists by using the Record Lookup element. If a matching contact doesn't exist, create a record for that user by using the Record Create element.

Usage

To create a single record with all field values from one sObject variable, or multiple records with all field values from an sObject collection, use Fast Create.

Note: At run time, the record isn't created until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

Create Salesforce Records from a Flow
Operators in Flow Record Filters
Customize What Happens When a Flow Fails
Define the Path That a Flow Takes
Flow Elements

Flow Record Delete Element

Deletes all Salesforce records that meet specified criteria.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Delete`	Identifies the object whose records you want to delete.
Filter Criteria	Narrows down the scope of records that the flow deletes.

Considerations for Defining Filter Criteria

- When you define multiple filters, the filter logic usually defaults to AND. However, if multiple filters have the same field selected and use the equals operator, the filters are combined with OR.

  For example, your filters check whether a case's Type equals Problem (1), Type equals Feature Request (2), and Escalated equals true (3). At runtime, those filters are combined to `(1 OR 2) AND 3`.

- The available filter operators depend on the data type of the selected fields. For details, see Operators in Flow Record Filters.

Usage

Warning:

- Be careful when testing flows that contain delete elements. Even if the flow is inactive, it triggers the delete operation.
- To prevent deleting records by mistake, be as specific in your filter criteria as possible.
- Records are deleted from your organization the moment the flow executes the delete element.
- Deleted records are sent to the Recycle Bin and remain there for 15 days before they are permanently deleted.
- Flows can delete records that are pending approval.

To delete a single record identified by the ID in one sObject variable, or delete multiple records identified by the IDs in an sObject collection, use Fast Delete.

Note: At run time, the record isn't deleted until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

Delete Salesforce Records from a Flow
Operators in Flow Record Filters
Customize What Happens When a Flow Fails
Define the Path That a Flow Takes
Flow Elements

Flow Record Lookup Element

Finds the first Salesforce record that meets specified criteria. Then assigns the record's field values to individual flow variables or individual fields on sObject variables.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Look up`	Identifies the object whose records you want to look up.
Filter criteria	Narrows down the scope of records that the flow looks up. Specify the filter criteria for selecting the record from the database. `Value` must be compatible with the selected field's data type. Tip: Make sure that your filter criteria sufficiently narrows the search. The Record Lookup element returns only the first record from the filtered results.
`Sort results by:`	Sorts the filtered results before storing records in the variable. If selected, also select the field that you want to sort the results by and the sort order. Only sortable fields are available.
`Assign the record's fields to variables to reference them in your flow.`	Select fields from the returned record, and assign the values to variables in the flow. The values must be compatible with each selected field.
`Assign null to the variable(s) if no records are found.`	Sets the variables to null if no records meet the filter criteria. By default, the variable's values are left unchanged.

Example: Use a Record Lookup element to:

- Input (or read) a bar code from a product tag. Use the code to find out the product name or description from the database.
- Look up item details to check your stock for availability.
- Look up a customer record to verify a caller's identity.

Considerations for Defining Filter Criteria

- When you define multiple filters, the filter logic usually defaults to AND. However, if multiple filters have the same field selected and use the equals operator, the filters are combined with OR.

  For example, your filters check whether a case's Type equals Problem (1), Type equals Feature Request (2), and Escalated equals true (3). At runtime, those filters are combined to `(1 OR 2) AND 3`.

- The available filter operators depend on the data type of the selected fields. For details, see Operators in Flow Record Filters.

Usage

Use a Fast Lookup element to find:

- A single record and store specified field values in an sObject variable
- Multiple records and store specified field values in an sObject collection

SEE ALSO:

- Pull Values from Salesforce Records into a Flow
- Operators in Flow Record Filters
- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Record Update Element

Finds all Salesforce records that meet specified criteria and updates them with individual field values that you specify.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Update`	Identifies the object whose records you want to update.
Filter criteria	Narrows down the scope of records that the flow updates. Important: Configure at least one filter, or the flow updates **all** the records for the object. `Value` must be compatible with the selected field's data type.
Update record fields ...	Identifies which fields to update on the records that meet the filter criteria, as well as the new values. The values must be compatible with each selected field.

Example: Automatically update Salesforce with data collected from customers, such as new addresses or product preferences.

Considerations for Defining Filter Criteria

- When you define multiple filters, the filter logic usually defaults to AND. However, if multiple filters have the same field selected and use the equals operator, the filters are combined with OR.

  For example, your filters check whether a case's Type equals Problem (1), Type equals Feature Request (2), and Escalated equals true (3). At runtime, those filters are combined to `(1 OR 2) AND 3`.

- The available filter operators depend on the data type of the selected fields. For details, see Operators in Flow Record Filters.

Usage

Use Fast Update to:

- Update a single record with all field values from an sObject variable
- Update multiple records with all field values from an sObject collection

Note: At run time, the record isn't updated until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

Update Salesforce Records from a Flow
Operators in Flow Record Filters
Customize What Happens When a Flow Fails
Define the Path That a Flow Takes
Flow Elements

Flow Quick Action Element

Calls an object-specific or global quick action that's already been configured in your organization. Only "Create," "Update," and "Log a Call" actions are supported.

The unique name for each object-specific action is prefixed with the object it's associated with. The unique name for each global action has no prefix.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Related Record ID`	Only for object-specific actions. The ID of the record from which the action executes. For example, the action creates a case that's associated with a given account. Assign the ID for that account to `Related Record ID`. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Input Parameter`	Varies for each action. The action layout determines which parameters are required. Required parameters appear by default and can't be removed. If a required field has a default or predefined value, that field is optional in object-specific and global actions in the flow. If you later remove the field's default or predefined value and you didn't set a value in the flow, the interview fails at run time. The value must be compatible with the parameter.

Example: Your organization has an object-specific action that creates a case record on an account. The flow calls that action at run time and uses input assignments to transfer data from the flow to the action.

Note: At run time, the record isn't created or updated until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Create Salesforce Records from a Flow
- Update Salesforce Records from a Flow
- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Post to Chatter Element

Posts a message to a specified feed, such as to a Chatter group or a case record. The message can contain mentions and topics, but only text posts are supported.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Inputs

Input Parameter	Description
`Community ID`	ID of a community to post to. Valid only if Salesforce Communities is enabled. Required if posting to a user or Chatter group that belongs to a Salesforce.com Community. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Message`	The text that you want to post. • To mention a user or group, enter *`@[reference]`*, where *`reference`* is the ID for the user or group that you want to mention. The reference can be a literal value, a merge field, or a flow resource. For example: `@[{!UserId}]`. • To add a topic, enter *`#[string]`*, where *`string`* is the topic that you want to add. For example: `#[Action Required]`. This parameter accepts **single-value resources of any type**. That value is treated as text and is limited to 10,000 characters.
`Target Name or ID`	Reference to the user, Chatter group, or record whose feed you want to post to. • To post to a user's feed, enter the user's `ID` or `Username`. For example: *`jsmith@salesforce.com`* • To post to a Chatter group, enter the group's `Name` or `ID`. For example: *`Entire Organization`* • To post to a record, enter the record's `ID`. For example: *`001D000000JWBDx`*

Input Parameter	Description
	This parameter accepts **single-value resources of any type**. That value is treated as text.
`Target Type`	Required only if `Target Name or ID` is set to a username or a Chatter group name. The type of feed that you want to post to. Valid values are: • *`User`*—If `Target Name or ID` is set to a user's `Username`, enter this value. • *`Group`*—If `Target Name or ID` is set to a Chatter group's `Name`, enter this value.
`Visibility`	Specifies whether this feed item is available to all users or internal users only. Valid only if Salesforce Communities is enabled. Valid values are: • *`allUsers`* • *`internalUsers`*

Outputs

Output Parameter	Description
`Feed Item ID`	Assigns the created post's ID to a resource in the flow. This parameter accepts any **single-value variables of type Text, Picklist, or Picklist (Multi-Select)**.

Usage

Note: At run time, the Chatter post isn't created until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Screen Element

Displays a screen to the user who is running the flow, which lets you display information to the user or collect information from the user.

IN THIS SECTION:

Flow Screen Element: General Info

Identifies which navigation buttons are available for a given screen, as well as whether help text is available to the flow user.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Flow Screen Element: User Input Fields

Lets users manually enter information. A *flow user input field* is a text box, long text area, number, currency, date, date/time, password, or checkbox in a Screen element.

Flow Screen Element: Choice Fields

Lets users select from a set of choices. A *flow choice field* is a radio button, drop-down list, multi-select checkbox, or multi-select picklist in a Screen element.

Flow Screen Element: Display Text Fields

Displays information to users while they're running the flow.

SEE ALSO:

Flow Elements

Define the Path That a Flow Takes

Flow Screen Element: General Info

Identifies which navigation buttons are available for a given screen, as well as whether help text is available to the flow user.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Navigation Options

Field	Description
Drop-down list	Determines which navigation buttons display for this screen. At run time, the system automatically determines which buttons are relevant, depending on whether any screens precede or follow the screen in the flow path. To restrict the screen from displaying either the **Previous** or the **Finish** button, use the drop-down list. • **No navigation restrictions**—(Default) The system displays all relevant navigation buttons on the screen. • **Don't show Previous button**—Select this option if revisiting the previous screen triggers an action that must not be repeated, such as a credit card transaction. • **Don't show Finish button**—Select this option if you need the user to go back to a previous screen to continue or complete the flow. For example, suppose the flow prompts the user to enter information to identify an existing contact. The flow then looks up the user-entered information in the database. If no matching contact is found, the flow displays a screen to tell the user to go back and try again. For that screen, select **Don't show Finish button**.
`Show Pause button`	Adds the **Pause** button to the navigation buttons for this screen, if `Let Users Pause Flows` is enabled in your organization's Workflow and Approvals settings. Once a user pauses a flow interview, only that user or an administrator can resume the interview. Note: Users can't resume paused flows from Lightning Experience, so we recommend removing the Pause button from flows that are distributed in Lightning Experience.

Field	Description
`Paused Message`	The message that's displayed to flow users when they pause a flow.

Help Text

Field	Description
Text box	Information for users to see when they click **Help for this form**. This field accepts **single-value resources of any type**. That value is treated as text.

Usage
If you allow users to pause interviews of this flow:

- Customize the interview label
- Enable `Let Users Pause Flows` in your organization's Process Automation Settings
- Add the Paused Flow Interviews component to the Home page layout for relevant users

SEE ALSO:

Flow Screen Element

Design Home Page Layouts

Flow Screen Element: User Input Fields

Lets users manually enter information. A *flow user input field* is a text box, long text area, number, currency, date, date/time, password, or checkbox in a Screen element.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Label`	User-friendly text that displays to the left of the field. To format the label, click .
`Unique Name`	A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Input Type`	Automatically populated based on the type of input field you selected.
`Default Value`	Pre-populated value for the input field. If the associated screen isn't executed, the stored value of the input field is always null. The data type of the default value must be compatible with the field's data type. For example, a checkbox's default value must be of type boolean.
`Scale`	Controls the number of digits to the right of the decimal point. Can't exceed 17. If you leave this field blank or set to zero, only whole numbers display when your flow runs. Available for only Currency and Number input fields.

Field	Description
`Required`	Forces users to enter a value before they can move on to the next screen.
`Validate`	Enables input validation on this field.
`Formula Expression`	Boolean formula expression that evaluates whether the user entered an acceptable value.
`Error Message`	Displays underneath the field if the user didn't enter an acceptable value. This field accepts **single-value resources of any type**. That value is treated as text.
Help Text	Adds next to the input field. In the text box, enter helpful information about this field. This field accepts **single-value resources of any type**. That value is treated as text.

SEE ALSO:

Flow Screen Element

Flow Screen Element: Choice Fields

Lets users select from a set of choices. A *flow choice field* is a radio button, drop-down list, multi-select checkbox, or multi-select picklist in a Screen element.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Label`	Displays to the left of the field. To format the label, click .
`Unique Name`	A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Value Data Type`	Controls which choices are available in Choice Settings. For example, if you choose Number you can't use a choice that has a data type of Text. You can't change the value data type of multi-select choice fields; only text is supported.
`Scale`	Controls the number of digits to the right of the decimal point. Can't exceed 17. If you leave this field blank or set to zero, only whole numbers display when your flow runs. Available for only Currency and Number choice fields.
`Required`	Forces users to identify a choice before they progress to the next screen.
`Default Value`	The choice that's preselected for the user. If the associated screen isn't executed, the stored value of the choice field is always null.
`Choice`	The choice options that the user can choose from. Select configured choices, dynamic record choices, or picklist choices. Note: You can't rearrange choices. The choice's data type must match Value Data Type.

Field	Description
`Help Text` box	Information that users see when they click  . This field accepts **single-value resources of any type**. That value is treated as text.

SEE ALSO:

- Limitations for Multi-Select Choice Fields
- Options for Choice Fields in Flow Screen Elements
- Flow Screen Element

Options for Choice Fields in Flow Screen Elements

A key part of configuring choice fields in a flow screen is selecting the choice options to display in that field. These options appear as the individual radio buttons or checkboxes or options in a drop-down list. Create choice options with at least one of these resources: choices, dynamic record choices, or picklist choices.

Dynamic record choices and picklist choices are easier to configure and don't require as much maintenance as choices. We recommend using a flow choice resource (otherwise known as stand-alone choices) only when you can't use the other two.

If you want the user to select...	Use this resource
From a set of filtered records	Dynamic Record Choice
From a set of values that correspond to an existing picklist or multi-select picklist field	Picklist Choice
Something that can't be generated from a record, picklist field, or multi-select picklist field	Choice

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

- Flow Screen Element: Choice Fields
- Flow Dynamic Record Choice Resource
- Flow Picklist Choice Resource
- Flow Choice Resource

Flow Screen Element: Display Text Fields

Displays information to users while they're running the flow.

Field	Description
`Unique Name`	A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Text box	The text to display to the flow user. Click to switch between the plain text editor and the rich text editor.Using the rich text editor saves the content as HTML. This field accepts **single-value resources of any type**. That value is treated as text.

Example: Welcome users to the flow or display terms and conditions.

SEE ALSO:

Flow Screen Element

Flow Send Email Element

Sends an email by manually specifying the subject, body, and recipients in the flow.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tip:

- Store the text for the email body in a text template.
- To use an email template that exists in your organization, call an email alert instead.

Specify at least one recipient for the email. You can use both email address parameters, so long as the combined number of addresses is five or fewer.

Field	Description
`Body`	Text for the body of the email. The email is treated as plain text; HTML formatting isn't respected. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Subject`	Text for the subject of the email. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Email Addresses (comma-separated)`	Optional. Recipients of the email. For the email to send successfully, enter a value for `Email Addresses (comma-separated)` or `Email Addresses (collection)`. You can use both parameters, so long as the combined number of email addresses is five or fewer. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Email Addresses (collection)`	Optional. Recipients of the email. For the email to send successfully, enter a value for `Email Addresses (comma-separated)` or `Email Addresses (collection)`. You can use both parameters, so long as the combined number of email addresses is five or fewer.

Field	Description
	This parameter accepts **collection variables of type Text**.
`Sender Address`	The organization-wide email address that's used to send the email. Required only if `Sender Type` is set to *`OrgWideEmailAddress`*. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Sender Type`	Optional. Email address used as the email's From and Reply-To addresses. Valid values are: • *`CurrentUser`*—Email address of the user running the flow. (Default) • *`DefaultWorkflowUser`*—Email address of the default workflow user. • *`OrgWideEmailAddress`*—The organization-wide email address that is specified in `Sender Address`.

Usage

Note: At run time, the email isn't sent until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Send Email from a Flow
- Flow Text Template Resource
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Step Element

Acts as a placeholder when you're not sure which element you need.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Usage

Steps aren't valid elements for active flows. As a flow admin, you can run a draft flow with Step elements in it. Before you activate the flow, replace the Step elements with other elements or delete them entirely.

Convert a Step element into a Screen element at any time by hovering your mouse over the step and clicking .

Note:

- If you convert a step that has multiple connectors into a Screen, all its connectors are deleted.

- Once a Step element has been converted, you can't use its original unique name.

SEE ALSO:

Flow Screen Element

Flow Elements

Define the Path That a Flow Takes

Flow Submit for Approval Element

Submits one Salesforce record for approval.

Tip: Before you begin, store the ID for the record that you want to submit for approval.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Inputs

Transfer data from the flow to the approval submission.

Input Parameter	Description
`Record ID`	The ID of the record that you want to submit for approval. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Next Approver IDs`	The ID of the user to be assigned the approval request when the approval process doesn't automatically assign the approver. This parameter accepts **collection variables of type Text** that include exactly one item.
`Approval Process Name or ID`	The unique name or ID of the specific approval process to which you want the record to be submitted. The process must have the same object type as the record you specified in `Record ID`. Required if `Skip Entry Criteria` is set to *`$GlobalConstant.True`*. If this parameter and `Submitter ID` aren't set, the flow succeeds only when: • The approver on submit is determined automatically, and • The user who launched the flow is an allowed initial submitter Make sure that: • The approver on submit is determined automatically, and • The initial submitters (for the approval processes related to this object) include all users who could launch this flow This parameter accepts **single-value resources of any type**. That value is treated as text.

Input Parameter	Description
`Skip Entry Criteria`	If set to *`$GlobalConstant.True`*, the record isn't evaluated against the entry criteria set on the process that is defined in `Approval Process Name or ID`. This parameter accepts any **single-value resource of type Boolean**.
`Submission Comments`	Text that you want to accompany the submission. Don't reference merge fields or formula expressions. Submission comments appear in the approval history for the specified record. This text also appears in the initial approval request email if the template uses the `{!ApprovalRequest.Comments}` merge field. This parameter accepts **single-value resources of any type**. That value is treated as text.
`Submitter ID`	The ID for the user who submitted the record for approval. The user receives notifications about responses to the approval request. The user must be one of the allowed submitters for the process. If you don't set this field, the user who launched the flow is the submitter. If a workflow rule triggers a flow that includes this element, the submitter is the user who triggered the workflow rule. Workflow rules can be triggered when a user creates or edits a record. When the record is approved or rejected, the user who launched the flow or triggered the workflow rule is notified. This parameter accepts **single-value resources of any type**. That value is treated as text.

Outputs

Transfer data from the approval request to the flow. Assignments occur when the approval request is created.

Optional Output Parameter	Description
`Instance ID`	The ID of the approval request that was submitted. This parameter accepts **single-value variables of type Text, Picklist, or Picklist (Multi-Select)**.
`Instance Status`	The status of the current approval request. Valid values are "Approved," "Rejected," "Removed," or "Pending". This parameter accepts **single-value variables of type Text, Picklist, or Picklist (Multi-Select)**.
`New Work Item IDs`	The IDs of the new items submitted to the approval request. There can be 0 or 1 approval processes. This parameter accepts **collection variables of type Text**.
`Next Approver IDs`	The IDs of the users who are assigned as the next approvers. This parameter accepts **collection variables of type Text**.
`Record ID`	The ID of the record that the flow submitted for approval. This parameter accepts **single-value variables of type Text, Picklist, or Picklist (Multi-Select)**.

Usage

Note: At run time, the approval request isn't created until the interview's transaction completes. Transactions complete either when the interview finishes, executes a Screen element, or executes a Wait element.

SEE ALSO:

- Flow Elements
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Subflow Element

Calls another flow in your organization. Use this element to reference modular flows and simplify the overall architecture of your flow.

A subflow element references another flow and calls that flow at run time. When a flow contains a subflow element, we call it the *master flow* to distinguish it from the *referenced flow*.

Tip: Create smaller flows that perform common tasks. For example, build utility flows to capture address and credit card information, and authorize a credit card purchase amount. Then call those flows as needed from multiple product-ordering flows.

Assign values to variables in the referenced flow. Variable assignments occur when the master flow calls the referenced flow at run time.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Column Header	Description
`Target`	Referenced flow's variable whose value you want to set. By default, this drop-down list contains the variables of the currently active version of the referenced flow. If the referenced flow has no active version, the drop-down list contains the variables of the *latest* version of the referenced flow.
`Source`	Master flow's resource or value to assign to the target. Options: • Select an existing flow resource, such as a variable, constant, or user input. • Select CREATE NEW to create a flow resource. • Manually enter a literal value or merge field. `Source`'s data type must be compatible with `Target`.

Assign values from the referenced flow's variables to the master flow's variables. Variable assignments occur when the referenced flow finishes running.

Column Header	Description
`Source`	Referenced flow's variable whose value you want to assign to the target. By default, this drop-down list contains the variables of the currently active version of the referenced flow. If the referenced flow has no active version, the drop-down list contains the variables of the *latest* version of the referenced flow.
`Target`	Master flow's variable whose value you want to set. `Target`'s data type must be compatible with `Source`.

Usage

At run time, the master flow calls the *active* version of each referenced flow by default. If a referenced flow has no active version, then the master flow calls the *latest* version of the referenced flow. To run only the latest version of each referenced flow:

- Open the master flow, and click **Run with Latest** in the button bar, or
- Append the URL for the master flow with `?latestSub=true`

Note: Only flow admins can run inactive flows. For other users, the flow fails at run time if a subflow element tries to call a flow with no active version.

SEE ALSO:

- Flow Elements
- View Inputs and Outputs of Other Referenced Flow Versions
- Customize What Happens When a Flow Fails
- Define the Path That a Flow Takes
- Cross-Object Field References in Flows

Flow Wait Element

Waits for one or more defined events to occur, which lets you automate processes that require a waiting period.

Define the events that the flow waits for before it proceeds.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Name`	The name appears on the connector that's associated with this event.
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Event Type`	Determines whether the flow is waiting for: • An absolute time (such as 3 days from now), or

Field	Description
	• A time relative to a date/time field on a Salesforce record (such as 3 days after an opportunity closes)
Event Conditions	Determines the exact event that the flow is waiting for. Parameters vary, based on the selected event type.
`Wait for this event only if additional conditions are met`	If selected, the flow waits for this event only when certain conditions are met.
Waiting Conditions	Determines which conditions must be true for the flow to wait for this event. Available only if `Wait for this event only if additional conditions are met` is selected. The flow waits for the event only if the waiting conditions evaluate to `true`. For details, see Define Flow Conditions on page 499.
Variable Assignments	Assigns the event's outputs to flow variables. Parameters vary, based on the selected event type.
`[Default Path]`	Determines what the flow does when all the events have unmet waiting conditions. If at least one event doesn't have waiting conditions, the default path is never executed. The name displays on the Wait element's default connector. Provide a custom name for this path by replacing the predefined value.

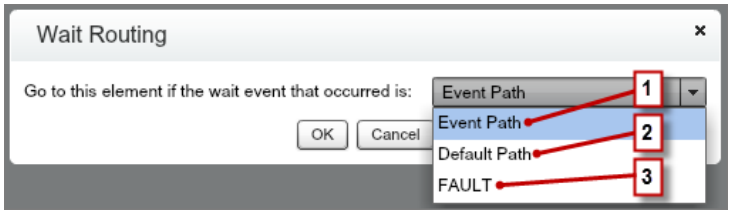
Usage

Note:

- Flows that contain Wait elements must be autolaunched. If a flow includes Wait elements and screens, choice, or dynamic choices, you can't activate or run it.
- Before you add a Wait element to your flow, understand the special behavior and limitations. See Limitations for Time-Based Flows on page 478 for details.

After you define your events, connect the Wait element to other elements on the canvas to indicate what the flow does when:

- Each event is the first to occur. One connector **(1)** is available for each event that's defined in the Wait element.
- There are no more events to wait for, because the waiting conditions for every event are unmet. One connector **(2)** is available for the Wait element's default path.
- An error occurs related to the Wait element. One connector **(3)** is available for the Wait element's fault path, and it's always labeled FAULT.



If the flow waits for multiple events, consider returning the flow path to the Wait element again so that the flow waits for the other events. If you return the flow path to the Wait element, consider using waiting conditions to control when the flow waits for each event. For an example, see Sample Flow That Waits for Many Events on page 587.

IN THIS SECTION:

SEE ALSO:

Customize What Happens When a Flow Fails
Define the Path That a Flow Takes
Cross-Object Field References in Flows
Flow Elements

What Are Waiting Conditions?

Each event that you define in a flow Wait element has optional *waiting conditions*. These conditions must be met for the flow interview to wait for that event at run time.

When an interview encounters a Wait element, it checks the waiting conditions for each event to determine which events to wait for. If the waiting conditions aren't met for an event, the interview doesn't wait for that event. If all events have unmet waiting conditions, the interview executes the default path.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tip: Add a default path if:

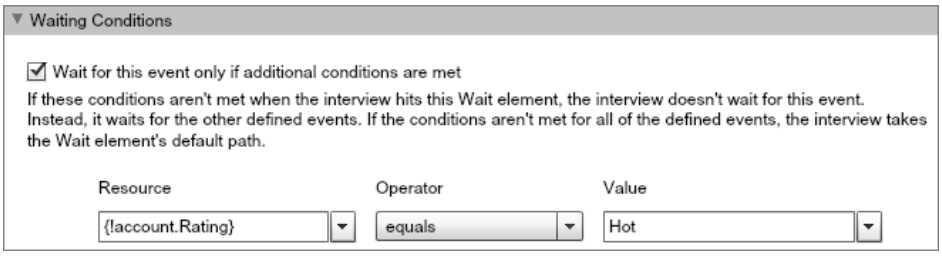
- All the events have waiting conditions set, and
- You want the flow to proceed when the waiting conditions for all events are met

Example: Here are two scenarios in which you would use waiting conditions.

- The flow waits for different events based on a field value on a given record.

  For example, send an email reminder to a contract's owner before the contract's end date. The date on which you send the email depends, however, on the rating of the contract's account. If the account is hot, send the email a month before the end date. If the account isn't hot, send the email two weeks before the end date.

  In this example, you would create two events. The event for hot accounts occurs 30 days before the contract's end date. Its waiting conditions would check if the `Rating` for the contract's account is equal to "Hot."



  The second event occurs 15 days before the contract's end date. Its waiting conditions would check if the `Rating` for the contract's account is not equal to "Hot."

When a flow interview executes the Wait element during run time, the interview checks the waiting conditions for each event. It only waits for the events whose waiting conditions are met. If the account is hot, the interview doesn't wait for the second event.

- The flow waits for multiple events to occur, such as to send periodic email reminders. For an example of this scenario, see Sample Flow That Waits for Many Events on page 587.

SEE ALSO:

Operators in Flow Record Filters
Flow Wait Element
Flow Event Types

Flow Resources

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Each *resource* represents a value that you can reference throughout the flow.

In the Cloud Flow Designer, the Explorer tab displays the resources that are available in the flow.

You can create some types of resources from the Resources tab by double-clicking them. Some resources, such as global constants and system variables, are automatically provided by the system. Other resources are provided by the system when you add an element to the flow. For example, when you add a Decision element to your flow, the system creates a resource for each outcome.

Which resources are available depend on the specific field you're setting. Oftentimes you can create resources from within that field by expanding the CREATE NEW section of its drop-down list.

Flow Resource	Description	Creatable from the Resources Tab
Choice	Represents an individual value that can be used in choice screen fields.	Yes
Collection Variable	Stores multiple updatable values that have the same data type, such as a group of email addresses.	Yes
Constant	Stores a fixed value.	Yes
Dynamic Record Choice	Represents a set of choices that's generated from an object's records.	Yes
Element	Any element that you add to the flow is available as a `Resource` with the `was visited` operator in outcome criteria. An element is considered visited if the element has already been executed in the flow interview. Any element that you add to the flow that supports a fault connector is available as a Boolean resource. If the element has already been successfully executed in the flow interview, the resource's value is `True`. If the element wasn't executed or was executed and resulted in an error, the resource's value is `False`.	

Flow Resource	Description	Creatable from the Resources Tab
Formula	Calculates a value by using other resources in your flow and Salesforce functions	Yes
Global Constant	Fixed, system-provided values, such as `EmptyString`, `True`, and `False`, that can be assigned as the values of flow resources.	
Global Variables	System-provided variables that reference information about the organization or running user, such as the user's ID or the API session ID. In Visual Workflow, global variables are available in only flow formulas.	
Outcome	If you add a Decision element to the flow, its outcomes are available as Boolean resources. If an outcome path has already been executed in the flow interview, the resource's value is `True`.	
Picklist Choice	Represents a set of choices that's generated from the values of a picklist or multi-select picklist field.	Yes
Picklist Values	System-provided values for picklist fields in sObject variables and sObject collection variables. Available for only Assignment and Decision elements.	
Screen Field	Any screen field that you add to the flow is available as a resource. The resource value depends on the type of screen field. The value for a screen input field is what the user enters. The value for a screen choice field is the stored value of the choice that the user selects. The value for a screen output field is the text that's displayed to the user.	
SObject Collection Variable	Stores updatable field values for one or more Salesforce records.	Yes
SObject Variable	Stores updatable field values for a Salesforce record.	Yes
System Variable	System-provided values about the running flow interview, such as `{!$Flow.CurrentDate}`, `{!$Flow.CurrentDateTime}`, and `{!$Flow.FaultMessage}`.	
Text Template	Stores formatted text.	Yes
Variable	Stores a value that can be updated as the flow executes.	Yes
Wait Event	If you add a Wait element to the flow, its events are available as Boolean resources. If an event's waiting conditions are met, the resource's value is `True`. If the event has no waiting conditions set, the resource's value is always `True`.	

SEE ALSO:

Cloud Flow Designer

Flow Choice Resource

Represents an individual value that can be used in choice screen fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Label`	A user-friendly label for the choice option.
`Unique Name`	A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this choice option from other resources.
`Value Data Type`	Controls which choice fields this choice can be used in. For example, you can't use a Text choice in a Currency radio button field.
`Scale`	Controls the number of digits to the right of the decimal point. Can't exceed 17. If you leave this field blank or set to zero, only whole numbers display when your flow runs. Available for only Currency and Number choices.
`Stored Value`	If the user selects this choice, the choice field has this value. If a user leaves a choice blank or unselected, its stored value is set to `null`.
`Show Input on Selection`	Displays a text box below the choice option. This option isn't available if the choice's data type is `Boolean`.

These fields appear only if you select `Show Input on Selection`.

Field	Description
`Label`	A user-friendly label for the text box.
`Required`	Forces users to enter a value in the text box before they can progress or finish the flow.
`Validate`	Evaluates whether the user entered an acceptable value.

These fields appear only if you select `Validate`.

Field	Description
`Formula`	Boolean formula expression that evaluates whether the user entered an acceptable value.
`Error Message`	Displays if the user didn't enter an acceptable value.

Example: If your flow asks users to choose a particular service level, create choices for Gold, Silver, and Bronze. In a screen, display the choices with a description of the features included. Then, in the same screen, let the user pick from a dropdown list.

SEE ALSO:

- Flow Resources
- Flow Screen Element: Choice Fields
- Options for Choice Fields in Flow Screen Elements
- Cross-Object Field References in Flows

Flow Collection Variable Resource

Stores multiple updatable values that have the same data type, such as a group of email addresses.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this collection variable from other resources.
`Data Type`	Determines the type of values that can be assigned to the collection variable.
`Input/Output Type`	Determines whether the collection variable can be accessed outside the flow. • Private—Can be assigned and used only within the flow • Input—Can be set at the start of the flow using Visualforce controllers, or subflow inputs • Output—Can be accessed from Visualforce controllers and other flows This field doesn't affect how variables are assigned or used within the same flow, for example, through these types of elements: Assignment, Record or Fast Create, Record or Fast Lookup, and Apex Plug-in. The default value of the field is `Private`. Warning: Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables

Field	Description
	from URL parameters, Visualforce controllers, subflows, and processes.

SEE ALSO:

- Flow Resources
- Flow sObject Collection Variable Resource
- Flow Loop Element
- Cross-Object Field References in Flows

Flow Constant Resource

A flow constant represents a fixed value. Unlike variables, this value can't change throughout a flow.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this constant from other resources.
`Data Type`	Determines the types of values that the constant can store.
`Value`	The constant's value. This value doesn't change throughout the flow.

SEE ALSO:

- Flow Resources

Flow Dynamic Record Choice Resource

Represents a set of choices that's generated from an object's records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this resource from other resources.
`Value Data Type`	Data type of the choice's stored value.

Field	Description
`Create a choice for each`	Determines the object whose records you want to use to generate the choices
Filter Criteria	Determines which records are included in the generated list of choices. If you don't apply any filters, a choice is generated for *every* record of the selected object. For example, to generate a list of all accounts in San Francisco, use filters to include only accounts whose Billing City is San Francisco.
`Choice Label`	Determines which field is used as the label for each generated choice. Select a field that enables users to differentiate between the generated choices. Tip: Make sure to choose a field that contains data. If the selected field has no value for a given record, the corresponding choice's label is blank at run time.
`Choice Stored Value`	Determines which field's value is stored when the user selects this choice at run time. `Value Data Type` determines the available options. By default, the stored value is null. The stored value is determined by the most recent user selection of a choice within the generated set.
`Sort results by`	Controls the order that the choices appear in. When `Sort results by` is selected, also select the field that you want to order the choices by. Then select which order the choices should appear in.
`Limit number of choices to`	Controls the number of options that appear in the screen field that uses this dynamic record choice. When `Limit number of choices to` is selected, also enter the maximum number (up to 200) of choices to include.
`Assign the record fields to variables...`	Takes field values from the record that the user chose and stores them in flow variables that you can reference later. Note: When a multi-select choice field uses a dynamic record choice, only values from the last record that the user selects are stored in the flow variables. If multiple multi-select choice fields on one screen use the same dynamic record choice, the variable assignments obey the first of those fields.

Example: In a support flow for a computer hardware manufacturer, users identify a product to find its latest updates. You create a dynamic record choice that displays all products whose product ID starts with a specific string of characters. However, the flow users are more likely to know the product's name than its ID, so for `Choice Label` select the field that contains the product

name. Elsewhere in the flow, you want to display the associated product ID and description. To do so, you assign the ID and Description field values from the user-selected record to flow variables.

SEE ALSO:

- Operators in Flow Record Filters
- Flow Screen Element: Choice Fields
- Options for Choice Fields in Flow Screen Elements
- Flow Resources

Flow Formula Resource

Calculates a value by using other resources in your flow and Salesforce functions.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this formula from other resources.
`Value Data Type`	The data type for the value calculated by the formula.
`Scale`	Controls the number of digits to the right of the decimal point. Can't exceed 17. If you leave this field blank or set to zero, only whole numbers display when your flow runs. Appears when `Value Data Type` is Number or Currency.
Formula	The formula expression that the flow evaluates at run time. The returned value must be compatible with `Value Data Type`. Note: Some formula operators are not supported in the Cloud Flow Designer.

SEE ALSO:

- Formula Operators and Functions
- Limitations for Flow Formulas
- Flow Resources
- Cross-Object Field References in Flows

Flow Global Constant Resource

Fixed, system-provided values, such as `EmptyString`, `True`, and `False`.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Global Constant	Supported Data Types
`{!$GlobalConstant.True}`	`Boolean`
`{!$GlobalConstant.False}`	`Boolean`
`{!$GlobalConstant.EmptyString}`	`Text`

Example: When you create a Boolean variable, `$GlobalConstant.True` and `$GlobalConstant.False` are supported. But when you create a Currency variable, no global constants are supported.

Null vs. Empty String

At run time, `{!$GlobalConstant.EmptyString}` and `null` are treated as separate, distinct values. For example:

- If you leave a text field or resource value blank, that value is `null` at run time. If you instead want the value to be treated as an empty string, set it to `{!$GlobalConstant.EmptyString}`.
- For flow conditions, use the `is null` operator to check whether a value is `null`. If the condition compares two text variables, make sure that their default values are correctly either set to `{!$GlobalConstant.EmptyString}` or left blank (`null`).

SEE ALSO:

Flow Resources

Global Variables in Visual Workflow

System-provided variables that reference information about the organization or running user, such as the user's ID or the API session ID. In Visual Workflow, global variables are available only in flow formulas.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: Use `{$!User.Id}` to easily access the ID of the user who's running the flow interview.

The following global variables are supported in flow formulas. If a value in the database has no value, the corresponding merge field returns a blank value. For example, if nobody has set a value for your organization's Country field, `{!$Organization.Country}` returns no value.

Global Variable	Description
`$Api`	References API URLs or the session ID. The following merge fields are available. • `Enterprise_Server_URL_`***`xxx`***—The Enterprise WSDL SOAP endpoint where *`xxx`* represents the version of the API. • `Partner_Server_URL_`***`xxx`***—The Partner WSDL SOAP endpoint where *`xxx`* represents the version of the API. • `Session_ID`

Global Variable	Description
`$Label`	References custom labels. This global variable appears only if custom labels have been created in your organization. The returned value depends on the language setting of the contextual user. The value returned is one of the following, in order of precedence: **1.** The local translation's text **2.** The packaged translation's text **3.** The master label's text
`$Organization`	References information about your company. Note: `{!$Organization.UiSkin}` returns one of these values. • `Theme1`—Obsolete Salesforce theme • `Theme2`—Salesforce theme used before Spring '10 • `Theme3`—Classic "Aloha" Salesforce theme, introduced in Spring '10 • `PortalDefault`—Salesforce Customer Portal theme • `Webstore`—Salesforce AppExchange theme
`$Profile`	References information from the current user's profile, such as license type or name. Tip: • Use profile names to reference standard profiles in `$Profile` merge fields. • Users don't need access to their profile information to run a flow that references these merge fields.
`$Setup`	References custom settings of type "hierarchy". This global variable appears only if hierarchy custom settings have been created in your organization. You can access custom settings of type "list" only in Apex. Hierarchical custom settings allow values at any of three different levels: • Organization—the default value for everyone • Profile—overrides the Organization value • User—overrides both Organization and Profile values Salesforce automatically determines the correct value for this custom setting field based on the running user's current context.
`$System`	`$System.OriginDateTime` represents the literal value of `1900-01-01 00:00:00`. Use this merge field to perform date/time offset calculations.
`$User`	References information about the user who's running the flow interview. For example, reference the user's ID or title. Tip: • The current user is the person who caused the flow to start. • When a flow is started because a Web-to-Case or Web-to-Lead process changed a record, the current user is the `Default Lead Owner` or `Default Case Owner`.

Global Variable	Description
	`$User.UITheme` and `$User.UIThemeDisplayed` identify the look and feel the running user sees on a given Salesforce page. The difference between the two variables is that `$User.UITheme` returns the look and feel the user is *supposed* to see, while `$User.UIThemeDisplayed` returns the look and feel the user *actually* sees. For example, a user may have the preference and permissions to see the Lightning Experience look and feel, but if they are using a browser that doesn't support that look and feel, for example, older versions of Internet Explorer, `$User.UIThemeDisplayed` returns a different value. These merge fields return one of the following values. • `Theme1`—Obsolete Salesforce theme • `Theme2`—Salesforce Classic 2005 user interface theme • `Theme3`—Salesforce Classic 2010 user interface theme • `Theme4d`—Modern "Lightning Experience" Salesforce theme • `Theme4t`—Salesforce1 mobile Salesforce theme • `PortalDefault`—Salesforce Customer Portal theme • `Webstore`—Salesforce AppExchange theme
`$UserRole`	References information about the current user's role, such as the role name or ID. Tip: • The current user is the person who caused the flow to start. • When a flow is started because a Web-to-Case or Web-to-Lead process changed a record, the current user is the `Default Lead Owner` or `Default Case Owner`.

Flow Picklist Choice Resource

Represents a set of choices that's generated from the values of a picklist or multi-select picklist field.

EDITIONS

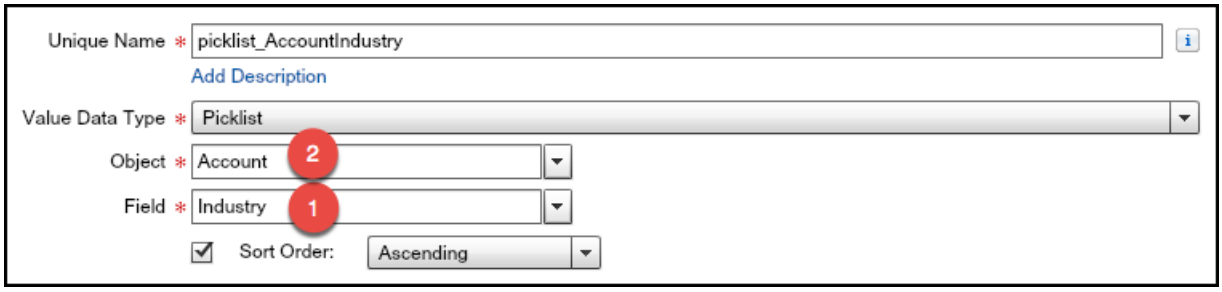
Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this resource from other resources.
`Value Data Type`	Data type of the choice's stored value.
`Object`	The object whose fields you want to select from.
`Field`	The picklist or multi-select picklist field to use to generate the list of choices.
`Sort Order`	Controls the order that the choices appear in.

Example: In a flow that simplifies the process of creating an account, users identify the company's industry.

Rather than creating one choice for each industry, you add a picklist choice to the flow and populate a drop-down list with it. When users run this flow, the picklist choice finds all the values in the database for the Industry picklist field **(1)** on the Account object **(2)**.



On top of being easier to configure than the stand-alone choice resource, picklist choices reduce maintenance. When someone adds new options to the Account Industry picklist, the flow automatically reflects those changes; you don't have to update the flow.

Limitations

Unlike with dynamic record choices, you can't:

Filter out any values that come back from the database.
The flow always displays every picklist value for that field—even if you're using record types to narrow down the picklist choices in page layouts.

Customize the label for each option.
The flow always displays the label for each picklist value.

Customize the stored value for each option.
The flow always stores the API value for each picklist value.

Picklists for Knowledge Articles aren't supported.

Labels and Values for Translated Fields

When a picklist field has been translated:

- Each choice's label uses the version of that picklist value in the running user's language
- Each choice's stored value uses the version of that picklist value in the org's default language

SEE ALSO:

Flow Screen Element: Choice Fields

Options for Choice Fields in Flow Screen Elements

Flow Resources

Flow sObject Collection Variable Resource

Stores updatable field values for one or more Salesforce records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this sObject collection variable from other resources.
`Input/Output Type`	Determines whether the sObject collection variable can be accessed outside the flow. • Private—Can be assigned and used only within the flow • Input—Can be set at the start of the flow using Visualforce controllers, or subflow inputs • Output—Can be accessed from Visualforce controllers and other flows The default value is `Private`. Warning: Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables from URL parameters, Visualforce controllers, subflows, and processes.
`Object Type`	Type of Salesforce records that the sObject collection represents in the flow.

Usage

After you populate the sObject collection, reference it to create, update, or delete records in the Salesforce database.

Examine every item in the collection by using a Loop element. When an item is being examined in the loop, the item's field values are copied into an sObject variable that you specify as the *loop variable*. If you want the loop to modify a collection item, such as to update an item's field values:

1. Configure the elements within the loop to update the loop variable.
2. Add the variable's field values to a separate collection.

You can add new items to the end of the collection (Assignment element) or replace all items in the collection (Fast Lookup element). However, you can't update existing collection items. To get around this limitation, have the loop iteratively add the contents of the loop variable to another collection. When the loop finishes, you can update the Salesforce records with values from the new collection.

SEE ALSO:

- Sample Flow That Loops Through a Collection
- Flow Loop Element
- Flow Resources

Flow sObject Variable Resource

Stores updatable field values for a Salesforce record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this sObject variable from other resources.
`Input/Output Type`	Determines whether the sObject variable can be accessed outside the flow. • Private—Can be assigned and used only within the flow • Input—Can be set at the start of the flow using Visualforce controllers, or subflow inputs • Output—Can be accessed from Visualforce controllers and other flows This field doesn't affect how variables are assigned or used within the same flow, for example, through these types of elements: Assignment, Record or Fast Create, Record or Fast Lookup, and Apex Plug-in. The default value of the field is `Private`. Warning: Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables from URL parameters, Visualforce controllers, subflows, and processes.
`Object Type`	Type of Salesforce record that the sObject variable represents in the flow.

Usage

When an sObject variable is created, its default value is `null`. Before you reference an sObject variable's values, make sure that the sObject variable has a value by using the `is null` operator in a Decision element.

SEE ALSO:

Flow Resources

System Variables in Flows

System-provided values about the running flow interview, such as `{!$Flow.CurrentDate}`, `{!$Flow.CurrentDateTime}`, and `{!$Flow.FaultMessage}`.

System Variable	Supported Resource Types	Description
`{!$Flow.CurrentDate}`	`Text`, `Date`, and `DateTime`	Date when the flow interview executes the element that references the system variable.
`{!$Flow.CurrentDateTime}`	`Text`, `Date`, and `DateTime`	Date and time when the flow interview executes the element that references the system variable.
`{!$Flow.FaultMessage}`	`Text`	System fault message that can help flow administrators troubleshoot run time issues.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: A flow is used only internally by call center personnel. For each flow element that interacts with the Salesforce database, a fault connector leads to a screen. A Display Text field on the screen displays the system fault message and instructs the flow user to provide that message to the IT department.

```
Sorry, but you can't read or update records at this time.
Please open a case with IT, and include the following error message:
{!$Flow.FaultMessage}
```

SEE ALSO:

Customize What Happens When a Flow Fails

Flow Resources

Flow Text Template Resource

Stores HTML-formatted text.

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this text template from other resources.
Text Template	The text for the template. Use HTML to format the text and merge fields to reference information from other resources.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example: You're designing a flow that registers people for an event. You create a text template that includes a registrant's name, address, and other information. Then you use the template in an email confirmation that the flow sends when it finishes.

SEE ALSO:

- Flow Resources
- Cross-Object Field References in Flows

Flow Variable Resource

Stores a value that can be updated as the flow executes.

Field	Description
`Unique Name`	The requirement for uniqueness applies only to elements within the current flow. Two elements can have the same unique name, provided they are used in different flows. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Description`	Helps you differentiate this variable from other resources.
`Data Type`	Determines the types of values that can be assigned to the variable.
`Scale`	Controls the number of digits to the right of the decimal point. Can't exceed 17. If you leave this field blank or set to zero, only whole numbers display when your flow runs. Appears only when the `Data Type` is set to `Number` or `Currency`.
`Input/Output Type`	Determines whether the variable can be accessed outside the flow. • Private—Can be assigned and used only within the flow • Input—Can be set at the start of the flow using Visualforce controllers, or subflow inputs

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
	• Output—Can be accessed from Visualforce controllers and other flows This field doesn't affect how variables are assigned or used within the same flow, for example, through these types of elements: Assignment, Record or Fast Create, Record or Fast Lookup, and Apex Plug-in. The default value of the field depends on the release or API version in which the variable is created: • `Private` for a variable created in Summer '12 and later or in API version 25.0 and later. • `Input and Output` for a variable created in Spring '12 and earlier or in API version 24.0. Warning: Disabling input or output access for an existing variable can break the functionality of applications and pages that call the flow and access the variable. For example, you can access variables from URL parameters, Visualforce controllers, subflows, and processes.
`Default Value`	Determines the variable value when the flow starts. If you leave this field blank, the value is `null`. Default values aren't available for Picklist and Picklist (Multi-Select) variables.

Usage

You can delete a variable at any time. Any variable assignments that use the deleted variable are set to `null`.

SEE ALSO:

- Flow Resources
- Flow Assignment Element
- Flow sObject Variable Resource
- Cross-Object Field References in Flows

Cross-Object Field References in Flows

When building a flow, you can reference fields for records that are related to the values that are stored in an sObject variable. To do so, manually enter the references.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Tips for Cross-Object Field References in Flows

Cross-object field values are valid wherever you can reference a flow resource or manually enter a value. Keep these implementation tips in mind when you use a cross-object field reference.

Cross-Object Field References in Flows: Simple Relationships

Most relationships are straightforward. For example, `Case.AccountId` links directly to the case's parent account. If you know that a field relationship ties your object to exactly one other object, use this syntax.

Cross-Object Field References in Flows: Polymorphic Relationships

Some fields have relationships to more than one object. We call these relationships *polymorphic*. For example, if you have queues enabled for cases, a case owner can be either a user or queue. If you're traversing from a case to its owner ID, add special syntax to identify which object you mean when you say "Owner".

Example Cross-Object Field References in Flows

This example demonstrates how to update a contract's owner to be the contract's account's owner.

Tips for Cross-Object Field References in Flows

Cross-object field values are valid wherever you can reference a flow resource or manually enter a value. Keep these implementation tips in mind when you use a cross-object field reference.

When you create an sObject variable to reference fields on related records from, store the ID for the first related record in the variable. For example, to reference an opportunity's contract, store `ContractId` in the sObject variable or add a value for `ContractId` by using an Assignment element.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Unsupported Relationships

The following relationships aren't supported in cross-object field references.

- `Lead.ConvertedAccount`
- `Lead.ConvertedContact`
- `Lead.ConvertedOpportunity`

Avoiding Null Values

If a flow interview encounters a null value at any point in the cross-object expression, the element containing the reference fails. The reference runs successfully if the last field value in the expression is `null`. For example, store a contact in `{!sObjContact}` and try to reference *`{!sObjContact}.Account.Name`*. The flow fails if `AccountId` on the stored contact is `null` (because there isn't an account to look at), but it succeeds if `Name` on the related account is `null`.

If an element contains a cross-object reference that fails and the element doesn't have a fault path defined, the entire interview fails. To avoid this situation, you can:

- Make the fields that you want to reference in the expression required in Salesforce. For example, for the expression *`{!sObjContact}.Account.Name`*, you could require `AccountId` on contact page layouts. Then, using another flow, find any records with null values for that field and update them.
- Determine whether each field that's referenced in the expression has a value by using the `wasSet` operator in a Decision element.

Cross-Object Field References and Org Limits

Cross-object field references in flows don't count against your org's limits for:

- Cross-object relationships per object
- DML operations per transaction

Cross-Object Field References in Flows: Simple Relationships

Most relationships are straightforward. For example, `Case.AccountId` links directly to the case's parent account. If you know that a field relationship ties your object to exactly one other object, use this syntax.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

To reference a field on a related record, use this syntax.

```
{!sObjectVariable.objectName1.objectName2.fieldName}
```

where:

- *sObjectVariable* is the unique name for the sObject variable that you want to start from.
- *objectName1* is the API name for an object that's related to sObjectVariable's object type. The API names for all custom objects end in __r.
- (Optional) *objectName2* is the API name for an object that's related to *objectName1*.

  Your expression must include at least one object name, but you can add more objects as needed.
- *fieldName* is the name for the field that you want to reference on the last object in the expression. The API names for all custom fields end in __c.

For example, *{!sOv_Contact.Account.Id}* references Id of the account that's related to the contact record represented by an sObject variable in the flow.

Cross-Object Field References in Flows: Polymorphic Relationships

Some fields have relationships to more than one object. We call these relationships *polymorphic*. For example, if you have queues enabled for cases, a case owner can be either a user or queue. If you're traversing from a case to its owner ID, add special syntax to identify which object you mean when you say "Owner".

To reference a field on a related record, use this syntax.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
{!sObjectVariable.polymorphicObjectName1:specificObjectName2.fieldName}
```

where:

- *sObjectVariable* is the unique name for the sObject variable that you want to start from.
- *polymorphicObject* is the API name for a polymorphic relationship for *sObjectVariable*'s object type.
- *specificObjectName* is the API name for the object that you want to select from the polymorphic relationship.
- *fieldName* is the name for the field that you want to reference on the last object in the expression. All custom field API names end in __c.

For example: {!sObj_Case.Owner:User.Id} references the ID of the user who owns the case, while {!sObj_Case.Owner:Queue.Id} references the ID of the queue who owns the case. You can always add the polymorphic reference after several traversals ({!sObj_Case.Account.Owner:User.Id}) or in the middle of a reference ({!sObj_Case.Owner:User.Manager.Id}).

Supported Polymorphic Relationships

Not every relationship is polymorphic, so we recommend using the polymorphic syntax only when you know that the field can link to multiple objects. The following relationships are supported.

- Case.Source
- FeedItem.CreatedBy

- *Object*.Owner

  Where *Object* lets you set Owner to either a user or a queue. Group.Owner and Queue.Owner aren't supported.

When you create an sObject variable to reference fields on related records from, store the ID for the first related record in the variable. For example, to reference an opportunity's contract, store ContractId in the sObject variable or add a value for ContractId by using an Assignment element.

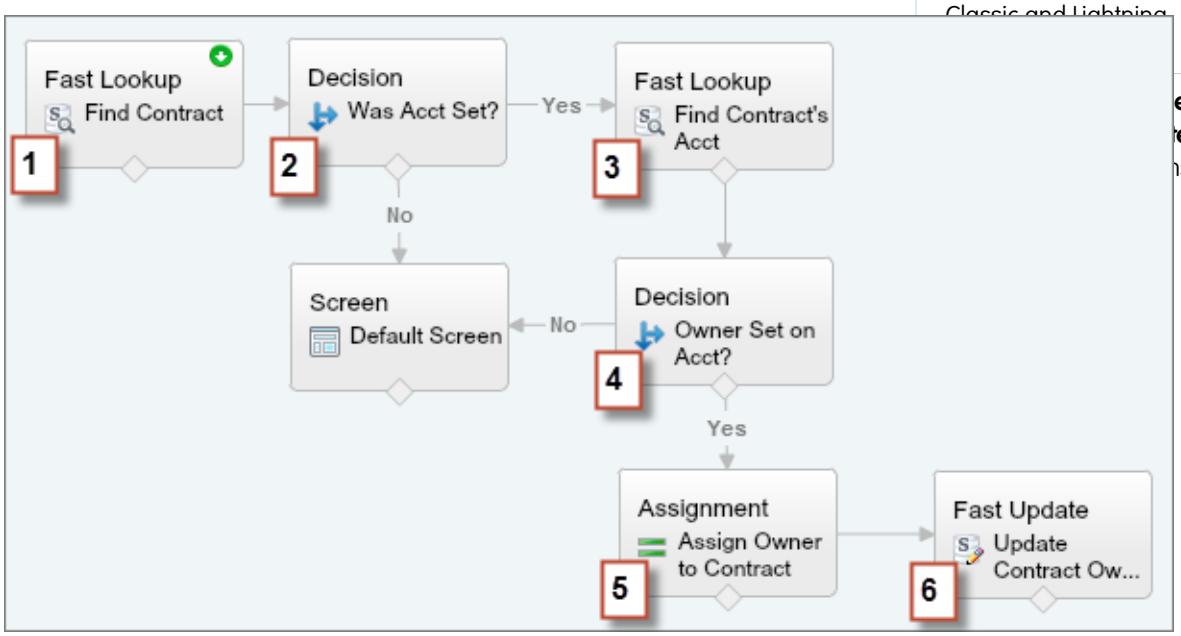
Example Cross-Object Field References in Flows

This example demonstrates how to update a contract's owner to be the contract's account's owner.

EDITIONS

Available in: both Salesforce Classic and Lightning

Example:



1. Use a Fast Lookup element to store the contract's fields, including AccountId, in an sObject variable called varContract.
2. Use a Decision element to verify that the value of AccountId was set in varContract.
3. Use a Fast Lookup to store the fields for the contract's account, including OwnerId, in another sObject variable called varAccount.
4. Use a Decision element to confirm that the value of OwnerId was set in varAccount.
5. Use an Assignment element to specify *{!varContract.Account.OwnerId}* as the value for {!varContract.OwnerId}.
6. Use a Fast Update element to write the values in varContract, including the updated OwnerId value, to the contract in Salesforce.

Flow Connectors

Connectors determine the available paths that a flow can take at run time. In the Cloud Flow Designer canvas, a connector looks like an arrow that points from one element to another.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Label	Example	Description
Unlabeled		Identifies which element to execute next.
`Decision outcome name`		Identifies which element to execute when the criteria of a Decision outcome are met.
`Wait event name`		Identifies which element to execute when an event that's defined in a Wait element occurs.
FAULT		Identifies which element to execute when the previous element results in an error.
Next element		Identifies the first element to execute for each iteration of a Loop element.
End of loop		Identifies which element to execute after a Loop element finishes iterating through a collection.

SEE ALSO:

Flow Elements

Flow Operators

Operators behave differently, depending on what you're configuring. In Assignment elements, operators let you change resource values. In flow conditions and record filters, operators let you evaluate information and narrow the scope of a flow operation.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Operators in Flow Assignment Elements

Use Assignment element operators to change the value of a selected resource.

Operators in Flow Conditions

Use condition operators to verify the value of a selected resource. Conditions are used in Decision elements and Wait elements.

Operators in Flow Record Filters

A flow record filter narrows the scope of records that the flow operates on. For example, use a record filter to update only the contacts that are associated with the Acme Wireless account. When you add a Record Update element, use the record filters to narrow the scope to just the contacts whose parent account is Acme Wireless.

Operators in Flow Assignment Elements

Use Assignment element operators to change the value of a selected resource.

Use this reference, organized by the data type that you select for Resource, to understand the supported operators.

- Boolean
- Collection
- Currency
- Date
- Date/Time
- Multi-Select Picklist
- Number
- Picklist
- sObject
- Text

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Boolean

Replace a boolean resource with a new value.

Operator	Description	Supported Data Types	Example
equals	What you enter or select for Value replaces the value of Variable.	Boolean	Before Assignment: `{!varBoolean}` is `false`. Assignment: `{!varBoolean}` **equals** `{!$GlobalConstant.True}` After Assignment: `{!varBoolean}` is `true`.

Collection

Replace the value of a collection variable or sObject collection variable (equals) or add an item to the end of the variable (add).

Operator	Description	Supported Data Types	Example
equals	What you enter or select for Value replaces the value of Variable.	Collection of the same data type or object type Text, Picklist, and Multi-Select Picklist data types are compatible with each other.	Before the Assignment: • `{!collText}` is `Yellow, Green, Blue` • `{!collPicklist}` is `Blue, Red, Orange` Assignment: `{!collText}` **equals** `{!collPicklist}`. After the Assignment: `{!collText}` is `Blue, Red, Orange`.

Operator	Description	Supported Data Types	Example
add	What you enter or select for Value is added as a new item at the end of the collection.	Variable of the same data type or sObject variable of the same object type Text, Picklist, and Multi-Select Picklist data types are compatible with each other.	Before the Assignment: • `{!collText}` is `Yellow, Green, Blue` • `{!varPicklist}` is `Red` Assignment: `{!collText}` **add** `{!varPicklist}`. After the Assignment: `{!collText}` is `Yellow, Green, Blue, Red`.

Currency and Number

Replace (equals), add to (add), or subtract from (subtract) the value of a currency or number resource.

Operator	Description	Supported Data Types	Example
equals	The number that you enter or select for Value replaces the value of Variable.	• Currency • Number	Before the Assignment: `{!varCurrency}` is `10`. Assignment: `{!varCurrency}` **equals** `7`. After the Assignment: `{!varCurrency}` is `7`.
add	The number that you enter or select for Value is added to the value of Variable.	• Currency • Number	Before the Assignment: `{!varCurrency}` is `10`. Assignment: `{!varCurrency}` **add** `7`. After the Assignment: `{!varCurrency}` is `17`.
subtract	The number that you enter or select for Value is subtracted from the value of Variable.	• Currency • Number	Before the Assignment: `{!varCurrency}` is `10`. Assignment: `{!varCurrency}` **subtract** `7`. After the Assignment: `{!varCurrency}` is `3`.

Date

Replace (equals), add to (add), or subtract from (subtract) the value of a date/time resource.

Operator	Description	Supported Data Types	Example
equals	The date that you enter or select for Value replaces the value of Variable.	• Date • Date/Time	Before the Assignment: `{!varDate}` is `1/16/2016`. Assignment: `{!varDate}` **equals** `1/15/2016`. After the Assignment: `{!varDate}` is `1/15/2016`.

Operator	Description	Supported Data Types	Example
add	Value is added, in days, to the selected Variable's value.	• Currency • Number	Before the Assignment: `{!varDate}` is `1/16/2016`. Assignment: `{!varDate}` **add** `7`. After the Assignment: `{!varDate}` is `1/23/2016`.
subtract	Value is subtracted, in days, from the selected Variable's value.	• Currency • Number	Before the Assignment: `{!varDate}` is `1/16/2016`. Assignment: `{!varDate}` **subtract** `7`. After the Assignment: `{!varDate}` is `1/9/2016`.

Date/Time

Replace a date/time resource with a new value (equals).

Operator	Description	Supported Data Types	Example
equals	The date that you enter or select for Value replaces the value of Variable.	• Date • Date/Time	Before the Assignment: `{!varDateTime}` is `1/16/2016 01:00`. Assignment: `{!varDateTime}` **equals** `1/16/2016 08:00`. After the Assignment: `{!varDateTime}` is `1/16/2016 08:00`.

Picklist

Replace a picklist resource with a new value (equals) or concatenate a value onto the original value (add).

Note: Before values are assigned or added to a picklist resource, they're converted into string values.

Operator	Description	Supported Data Types	Example
equals	What you enter or select for Value replaces the value of the selected picklist.	• Boolean • Currency • Date • Date/Time • Multi-Select Picklist • Number • Picklist • Text	Before the Assignment: `{!varPicklist}` is `Blue`. Assignment: `{!varPicklist}` **equals** `Yellow`. After the Assignment: `{!varPicklist}` is `Yellow`.

Operator	Description	Supported Data Types	Example
add	What you enter or select for Value is added to the end of the selected picklist.	• Boolean • Currency • Date • Date/Time • Multi-Select Picklist • Number • Picklist • Text	Before the Assignment: `{!varPicklist}` is `Blue`. Assignment: `{!varPicklist}` **add** `-green`. After the Assignment: `{!varPicklist}` is `Blue-green`.

Multi-Select Picklist

Replace a multi-select picklist resource with a new value (equals), concatenate a value onto the original value (add), or add a selection to the resource (add item).

Note: Before values are assigned or added to a multi-select picklist resource, they're converted into string values.

Operator	Description	Supported Data Types	Example
equal	What you enter or select for Value replaces the value of the selected multi-select picklist.	• Boolean • Collection • Currency • Date • Date/Time • Multi-Select Picklist • Number • Picklist • Text	Before the Assignment: `{!varMSP}` is `Blue`. Assignment: `{!varMSP}` **equals** `Yellow`. After the Assignment: `{!varMSP}` is `Yellow`.
add	What you enter or select for Value is added to the last item selected in the multi-select picklist. It doesn't create a selection. Easily add items to a multi-select picklist resource by using the "add item" operator. To add semi-colon-delimited items to a multi-select picklist variable with the "add" operator, always add	• Boolean • Currency • Date • Date/Time • Multi-Select Picklist • Number • Picklist • Text	Before the Assignment: `{!varMSP}` is `Blue; Green`. This value includes two separate selections. Assignment: `{!varMSP}` **add** `Yellow`. After the Assignment: `{!varMSP}` is `Blue; GreenYellow`. This value includes two separate selections.

Operator	Description	Supported Data Types	Example
	a single space after the semi-colon and don't include a space before the semi-colon. This way, you can compare the variable's values to the values of a multi-select picklist field from the Salesforce database. For example: `; Yellow`		
add item	What you enter or select for Value is added as a new selection to the end of the multi-select picklist. The Assignment automatically adds "; " before the new item. That way, Salesforce reads it as a separate item selected by the multi-select picklist.	• Boolean • Currency • Date • Date/Time • Multi-Select Picklist • Number • Picklist • Text	Before the Assignment: `{!varMSP}` is `Blue; Green`. Assignment: `{!varMSP}` **add** item `Yellow`. After the Assignment: `{!varMSP}` is `Blue; Green; Yellow`. This value includes three separate selections.

sObject

Replace an sObject variable with a new value (equals).

Operator	Description	Supported Data Types	Example
equals	The sObject that you select for Value replaces the value of Variable.	sObject with the same object type	Before the Assignment: • `{!account1}` contains field values for the Acme Wireless account • `{!account2}` contains field values for the Global Media account Assignment: `{!account1}` **equals** `{!account2}`. After the Assignment: both `{!account1}` and `{!account2}` contain the field values for the Global Media account.

Text

Replace a text resource with a new value (equals) or concatenate a value onto the end of the original value (add).

Note: Before values are assigned or added to a text resource, they're converted into string values.

Operator	Description	Supported Data Types	Example
equals	The text that you enter or select for Value replaces the value of Variable.	• Boolean • Currency • Date • Date/Time • Number • Multi-select picklist • Picklist • Text	Before the Assignment: `{!varText}` is `Blue`. Assignment: `{!varText}` **equals** `Yellow`. After the Assignment: `{!varText}` is `Yellow`.
add	The text that you enter or select for Value is added to the end of Variable.	• Boolean • Currency • Date • Date/Time • Number • Multi-select picklist • Picklist • Text	Before the Assignment: `{!varText}` is `Blue`. Assignment: `{!varText}` **add** `Yellow`. After the Assignment: `{!varText}` is `BlueYellow`.

Operators in Flow Conditions

Use condition operators to verify the value of a selected resource. Conditions are used in Decision elements and Wait elements.

Use this reference, divided up by the data type that you select for Resource, to understand the supported operators.

- Boolean
- Choice
- Collection
- Currency
- Date
- Date/Time
- Multi-Select Picklist
- Number
- Picklist
- sObject
- Text

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Boolean

Check whether a Boolean resource's value matches another value or resource.

Operator	True if...	Supported Data Types
does not equal	The value of the selected Resource doesn't match what you enter or select for Value.	Boolean
equals	The value of the selected Resource matches what you enter or select for Value. An outcome resolves to true if the flow interview took that outcome. A wait event resolves to true if all of the waiting conditions for that event are met.	Boolean
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once.	Boolean
was visited	The selected Resource is an element in the flow, and it has been visited during the flow interview.	Boolean

Choice

Every choice resource has a data type and obeys the operator rules for that data type. However, choice resources support one extra operator that other resources don't, no matter what their data type is.

Operator	True if...	Supported Data Types
was selected	A user selected that choice or dynamic record choice in a screen choice input field. If your flow references the same choice option in multiple screens, was selected always evaluates to the most recent screen that the flow visited. If your flow references the same choice option with a user input in more than one place on the same screen, this operator always evaluates the first usage in the screen.	Boolean

Collection

Check whether a Collection resource's value contains or matches another value or resource.

Operator	True if...	Supported Data Types
contains	An item in the collection that's selected for Resource contains the exact same value as Value	Varies If the resource is an sObject collection variable, only sObject resources with the same object type are supported.

Operator	True if...	Supported Data Types
		Otherwise, only resources with the same data type are supported.
does not equal	The collection that's selected for Resource doesn't match the collection that's selected for Value Two sObject collection variables are unequal if they include different fields or if the fields have different values.	Collection If the resource is an sObject collection variable, only sObject collection variables with the same object type are supported. Otherwise, only collection variables with the same data type are supported.
equals	The collection that's selected for Resource matches the collection that's selected for Value Two sObject collection variables are equal if they include the same fields and those fields have the same values.	Collection If the resource is an sObject collection variable, only sObject collection variables with the same object type are supported. Otherwise, only collection variables with the same data type are supported.
is null	The collection that's selected for resource isn't populated with any values	Boolean

Currency and Number

Check whether a Currency or Number resource's value matches, is larger than, or is smaller than another value or resource.

Operator	True if...	Supported Data Types
does not equal	The value for Resource doesn't match what's entered or selected for Value	• Currency • Number
equals	The value for Resource matches what's entered or selected for Value	• Currency • Number

Operator	True if...	Supported Data Types
greater than	The value of the Resource is larger than what's entered or selected for Value	• Currency • Number
greater than or equal	The value of the Resource is larger than what's entered or selected for Value or is the same	• Currency • Number
less than	The value of the Resource is smaller than what's entered or selected for Value	• Currency • Number
less than or equal	The value of the Resource is smaller than what's entered or selected for Value or is the same	• Currency • Number
is null	Resource isn't populated with a value	Boolean
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once	Boolean

Date and Date/Time

Check whether a Date or Date/Time resource's value matches, is before, or is after another value or resource.

Operator	True if...	Supported Data Types
does not equal	The value for Resource doesn't match what's entered or selected for Value	• Date • Date/Time
equals	The value for Resource matches what's entered or selected for Value	• Date • Date/Time
greater than	The value of the Resource is a later date or time than what's entered or selected for Value	• Date • Date/Time
greater than or equal	The value of the Resource is a later date or time than what's entered or selected for Value or is the same date or time	• Date • Date/Time
less than	The value of the Resource is an earlier date or time than what's entered or selected for Value	• Date • Date/Time
less than or equal	The value of the Resource is an earlier date or time than what's entered or selected for Value or is the same date or time	• Date • Date/Time

Operator	True if...	Supported Data Types
is null	Resource isn't populated with a value	Boolean
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once	Boolean

Picklist

Check whether a Picklist resource's value matches or contains another value or resource.

Note: These operators treat the resource's value as a text value.

Operator	True if...	Supported Data Types
contains	The value for Resource contains what's entered or selected for Value For example, if the value of `{!varPicklist}` is `yellow-green`, the condition `{!varPicklist}` **contains** `green` evaluates to true.	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
does not equal	The value for Resource doesn't match what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
equals	The value for Resource matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number

Operator	True if...	Supported Data Types
		• Picklist • Text
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once	Boolean

Multi-Select Picklist

Check whether a multi-select picklist resource's value matches or contains another value or resource.

Note: These operators treat the resource's value as a text value. If the resource's value includes multiple items, the operators treat the value as one string that happens to include semi-colons. It doesn't treat each selection as a different value. For example, the operators treat `red; blue; green` as a single value rather than three separate values.

Operator	True if...	Supported Data Types
contains	The value for Resource contains what's entered or selected for Value Tip: When you use this operator for a multi-select picklist resource, be aware of the values that a user can enter. If you want to check that a specific value is included and that value is also included as part of another value, create a flow formula resource that uses the INCLUDES function. For example, your organization has a Color multi-select picklist value. Among the possible values are "green" and "yellow-green". If both "green" and "yellow-green" are acceptable values, use the contains operator in a flow condition. If only "green" is an acceptable value, create a formula that uses the INCLUDES() function.	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
does not equal	The value for Resource doesn't match what's entered or selected for Value Note: Order matters. If you aren't sure which order the values that you're checking for will appear in, use the INCLUDES() function in a flow formula. For example, if you compare "red; blue; green" to "blue; green; red" using the does not equal operator, that condition resolves to true.	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
equals	The value for Resource exactly matches what's entered or selected for Value Note: Order matters. If you aren't sure which order the values that you're checking for will appear in, use the INCLUDES() function in a flow formula. For example, if you compare "red; blue; green" to "blue; green; red" using the equals operator, that condition will resolve to false.	• Boolean • Currency • Date • Date/Time

Operator	True if...	Supported Data Types
		• Multi-select Picklist • Number • Picklist • Text
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once	Boolean

s0bject

Check whether an sObject resource's value matches another value or resource.

Operator	True if...	Supported Data Types
does not equal	The value for Resource doesn't match what's entered or selected for Value	sObject with the same object type
equals	The value for Resource matches what's entered or selected for Value	sObject with the same object type
is null	Resource isn't populated with a value	Boolean

Text

Check whether a Text resource's value matches, contains, ends with, or starts with another value or resource.

Note: Before values are compared to a text resource, they're converted into string values.

Operator	True if...	Supported Data Types
contains	The value for Resource contains what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
does not equal	The value for Resource doesn't match what's entered or selected for Value	• Boolean

Operator	True if...	Supported Data Types
		• Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
equals	The value for Resource matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
ends with	The end of the value for Resource matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
is null	Resource isn't populated with a value	Boolean
starts with	The beginning of the value for Resource matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text

Operator	True if...	Supported Data Types
was set	The value for Resource is a field in an sObject variable, and that field has been populated with a value in the flow at least once	Boolean

Operators in Flow Record Filters

A flow record filter narrows the scope of records that the flow operates on. For example, use a record filter to update only the contacts that are associated with the Acme Wireless account. When you add a Record Update element, use the record filters to narrow the scope to just the contacts whose parent account is Acme Wireless.

Use this reference, organized by the data type of the field that you select, to understand the supported operators.

- Address Fields
- Autonumber Fields
- Checkbox Fields
- Currency Fields
- Date Fields
- Date/Time Fields
- Email Fields
- Encrypted Text Fields
- External Lookup Relationship Fields
- Fax Fields
- Lookup Relationship Fields
- Multi-Select Picklist Fields
- Number Fields
- Parent Fields
- Percent Fields
- Phone Fields
- Picklist Fields
- Text Fields
- Text Area (Long) Fields
- Text Area (Rich) Fields
- URL Fields

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Checkbox Fields

When you select a checkbox field under Field, these operators are available.

Operator	Filters to records where the selected field's value ...	Supported Data Types
does not equal	Doesn't match what you enter or select for Value	Boolean
equals	Matches what you enter or select for Value	Boolean
is null	Hasn't been populated with a value yet (if you select True for Value)	Boolean

Tip: Flow treats `null` as a different value than `false`. If you filter for records whose checkbox field is null, no records are returned.

Currency, Number, and Percent Fields

When you select a currency, number, or percent field under Field, these operators are available.

Operator	Filters to records where the selected field's value ...	Supported Data Types
does not equal	Doesn't match what's entered or selected for Value	• Currency • Number
equals	Matches what's entered or selected for Value	• Currency • Number
greater than	Is larger than what's entered or selected for Value	• Currency • Number
greater than or equal	Is larger than what's entered or selected for Value or is the same	• Currency • Number
is null	Hasn't been populated with a value yet (if you select True for Value)	Boolean
less than	Is smaller than what's entered or selected for Value	• Currency • Number
less than or equal	Is smaller than what's entered or selected for Value or is the same.	• Currency • Number

Date and Date/Time

When you select a date or date/time field under Field, these operators are available.

Operator	Filters to records where the selected field's value ...	Supported Data Types
does not equal	The value for Resource doesn't match what's entered or selected for Value	• Date • Date/Time
equals	The value for Resource matches what's entered or selected for Value	• Date • Date/Time
greater than	The value of the Resource is a later date or time than what's entered or selected for Value	• Date • Date/Time
greater than or equal	The value of the Resource is a later date or time than what's entered or selected for Value or is the same date or time	• Date • Date/Time
is null	Hasn't been populated with a value yet (if you select True for Value)	Boolean
less than	The value of the Resource is an earlier date or time than what's entered or selected for Value	• Date • Date/Time
less than or equal	The value of the Resource is an earlier date or time than what's entered or selected for Value or is the same date or time	• Date • Date/Time

Picklist and Text Fields

When you select a picklist or text field under Field, these operators are available.

Operator	Filters to records where the selected field's value ...	Supported Data Types
contains	Contains what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
does not equal	Doesn't match what's entered or selected for Value	• Boolean • Currency • Date

Operator	Filters to records where the selected field's value ...	Supported Data Types
		• Date/Time • Multi-select Picklist • Number • Picklist • Text
equals	Matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
ends with	Ends with what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
is null	Hasn't been populated with a value yet (if you select True for Value)	Boolean
starts with	Begins with what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text

Multi-Select Picklist Fields

When you select a multi-select picklist field under Field, these operators are available.

Tip: Be careful when using these operators to filter records based on a multi-select picklist field. Even if two resources have the same items in a multi-select picklist, they can be mismatched if these cases differ.

- The spacing before or after the semi-colon. For example, one resource's value is "red; green; blue" and the other's value is "red;green;blue"
- The order of the items. For example, one resource's value is "red; green; blue" and the other's value is "red; blue; green"

For best results, use the INCLUDES function in a flow formula.

Operator	Filters to records where the selected field's value ...	Supported Data Types
does not equal	Doesn't match what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
equals	Matches what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
ends with	Ends with what's entered or selected for Value	• Boolean • Currency • Date • Date/Time • Multi-select Picklist • Number • Picklist • Text
is null	Hasn't been populated with a value yet (if you select True for Value)	Boolean
starts with	Begins with what's entered or selected for Value	• Boolean • Currency

Operator	Filters to records where the selected field's value ...	Supported Data Types
		• Date • Date/Time • Multi-select Picklist • Number • Picklist • Text

Flow Event Types

`Event Type` drives the fields that you use to define an event in a flow Wait element. The available event types are both alarms, which consist of a date/time value—the base time—and an optional offset from that time.

The *base time*, which is always required, is the date/time value from which the alarm is based. If there's no offset for the alarm, the alarm is set to the exact value of the base time. The base time can be composed of one or multiple fields, based on the event type that you choose.

The *offset*, which is optional, is the amount of time before or after the base time at which the alarm occurs. An offset is always composed of two fields: `Offset Number` and `Offset Unit`. For example, if you want your alarm to occur three days after the base time, the number is `3` and the unit is `Days`.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Absolute Time Alarms

An *absolute time alarm* waits for a defined time that's based off an absolute date/time value. For example, you can use this event type in a Wait element to do something a day after the flow interview starts to wait.

Relative Time Alarms

A *relative time alarm* waits for a defined time that's based off a date/time field on a record. For example, you can use this event type to do something three days before a contract ends.

SEE ALSO:

Flow Wait Element

Absolute Time Alarms

An *absolute time alarm* waits for a defined time that's based off an absolute date/time value. For example, you can use this event type in a Wait element to do something a day after the flow interview starts to wait.

When you configure a Wait element in a flow:

- Define what the flow is waiting for
- Assign information from the event after it occurs to flow variables

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Event Conditions

The following parameters are available to define events with an `Event Type` of Alarm: Absolute Time.

Parameter	Description	Example
`Base Time`	A date/time value. If you enter values for `Offset Number` and `Offset Unit`, this field value is the base for the offset. You can manually enter a date/time value or reference a merge field or flow resource.	`{!$Flow.CurrentDate}`
`Offset Number`	Optional. The number of days or hours to offset `Base Time`. Required if you set a value for `Offset Unit`.The value must be a manually entered integer. You can't use a merge field or flow resource for this value. To set the alarm to occur before `Base Time`, use a negative number. To set the alarm to occur after `Base Time`, use a positive number.	*-3*
`Offset Unit`	Optional. The unit to offset `Base Time`. Required if you set a value for `Offset Number`. Manually enter *`Days`* or *`Hours`*. You can't use a merge field or flow resource for this value.	*Days*

For an example of a flow that waits for an absolute time alarm, see Sample Flow That Waits for a Single Event.

Event Outputs

Reference information from the event in your flow by assigning its outputs to flow variables.

Parameter	Description	Example
`Base Time`	The actual time at which the event occurred and the flow interview resumed.	`11/26/2014 10:12 AM`
`Event Delivery Status`	The status of the event when the flow interview resumed. After an event occurs, Salesforce delivers the event to the flow that's waiting for it, so that the flow knows to resume. Valid values are: • Delivered: The event was successfully delivered. • Invalid: An error occurred during delivery, but the flow successfully resumed.	`Delivered`

SEE ALSO:

Flow Event Types

Relative Time Alarms

A *relative time alarm* waits for a defined time that's based off a date/time field on a record. For example, you can use this event type to do something three days before a contract ends.

When you configure a Wait element in a flow:

- Define what the flow is waiting for
- Assign information from the event after it occurs to flow variables

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Event Conditions

The following parameters are available to define events with an `Event Type` of Alarm: Relative Time.

Parameter	Description	Example
`Object Type`	The API name of the object whose field you want to base the alarm on. See Supported Objects. on page 581 You must manually enter a string. You can't use a merge field or flow resource for this value.	*`Contract`*
`Base Date/Time Field`	The API name for a date or date/time field on the specified object. If you enter values for `Offset Number` and `Offset Unit`, this field value is the base for the offset. Manually enter a string.	*`EndDate`*
`Record ID`	ID of the record that the alarm is based on. The record's object type must match `Object Type`. You can enter a string, merge field, or flow resource.	*`{!ContractId}`*
`Offset Number`	Optional. The number of days or hours to offset `Base Date/Time Field`. Required if you set a value for `Offset Unit`. The value must be a manually entered integer. You can't use a merge field or flow resource for this value. To set the alarm to occur before `Base Date/Time Field`, use a negative number. To set the alarm to occur after `Base Date/Time Field`, use a positive number.	*`-3`*
`Offset Unit`	Optional. The unit to offset `Base Date/Time Field`. Required if you set a value for `Offset Number`. Manually enter *`Days`* or *`Hours`*. You can't use a merge field or flow resource for this value.	*`Days`*

For examples of flows that wait for relative time alarms, see Sample Flow That Waits for Only the First Event or Sample Flow That Waits for Many Events.

Event Outputs

Reference information from the event in your flow by assigning its outputs to flow variables.

Parameter	Description	Example
`Base Time`	The actual time at which the event occurred and the flow interview resumed.	`11/26/2014 10:12 AM`
`Event Delivery Status`	The status of the event when the flow interview resumed. After an event occurs, Salesforce delivers the event to the flow that's waiting for it, so that the flow knows to resume. Valid values are: • Delivered: The event was successfully delivered. • Invalid: An error occurred during delivery, but the flow successfully resumed.	`Delivered`

Supported Objects

You can create a relative time alarm for any custom object or any of the following standard objects.

- `Account`
- `Asset`
- `Campaign`
- `CampaignMember`
- `Case`
- `CaseComment`
- `Certification`
- `CertificationDef`
- `CertificationSectionDef`
- `CertificationStep`
- `CertificationStepDef`
- `Contact`
- `Contract`
- `ContractLineItem`
- `DandBCompany`
- `DuplicateRecordItem`
- `DuplicateRecordSet`
- `EmailMessage`
- `Entitlement`
- `EntitlementContact`
- `EnvironmentHubMember`
- `EnvironmentHubMemberRel`
- `Event`
- `ExternalEventMapping`

- FeedItem
- Goal
- GoalLink
- Idea
- IdentityProvEventLog
- Lead
- LiveAgentSession
- LiveChatTranscript
- LiveChatTranscriptEvent
- LiveChatTranscriptSkill
- Macro
- MacroAction
- MacroInstruction
- Metric
- MobileDeviceCommand
- Opportunity
- OpportunityLineItem
- OpportunitySplit
- OpportunityTeamMember
- Order
- OrderItem
- Organization
- PersonAccount
- Product2
- ProfileSkill
- ProfileSkillEndorsement
- ProfileSkillUser
- Question
- QuickText
- Quote
- QuoteLineItem
- Reply
- SOSSession
- SOSSessionActivity
- ServiceContract
- SignupRequest
- Site
- SocialPersona
- SocialPost
- Solution

- SsoUserMapping
- StreamingChannel
- Task
- UsageEntitlementPeriod
- User
- UserLicense
- UserProvisioningRequest
- WorkBadge
- WorkBadgeDefinition
- WorkCoaching
- WorkFeedback
- WorkFeedbackQuestion
- WorkFeedbackQuestionSet
- WorkFeedbackRequest
- WorkFeedbackTemplate
- WorkGoal
- WorkPerformanceCycle
- WorkReward
- WorkRewardFund
- WorkRewardFundType
- WorkThanks
- WorkUpgradeAction
- WorkUpgradeCustomer
- WorkUpgradeUser
- *articleType*_kav

SEE ALSO:

Flow Event Types

Flow Types

A flow or flow version's type determines which elements and resources are supported, as well as the ways that the flow can be distributed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Standard Flow Types

The following flow types are supported in the Cloud Flow Designer.

Type	Description
Flow	Requires user interaction, because it has one or more screens, steps, choices, or dynamic choices. This flow type doesn't support wait elements.

Type	Description
	A flow can be implemented with a custom button, custom link, direct URL, Visualforce page, or Salesforce1 action.
Autolaunched Flow	Doesn't require user interaction. This flow type doesn't support screens, steps, choices, or dynamic choices. An autolaunched flow can be implemented any way that a flow can, as well as with a process action, workflow action (pilot), or Apex code.
User Provisioning Flow	Provisions users for third-party services. A user provisioning flow can only be implemented by associating it with a connected app when running the User Provisioning Wizard. Provisions users for third-party services. For example, use this flow type to customize the user provisioning configuration for a connected app to link Salesforce users with their Google Apps accounts.

Other Flow Types

Not all flow types are supported in the Cloud Flow Designer. Some flow types are used in other parts of Salesforce. You can't create or edit these flows in the Cloud Flow Designer, so you don't see them in the list of flows. However, the Paused and Waiting Interviews list on the flow management page can display interviews with one of these types.

For example, when you run a process (from the Process Builder), a flow interview is created. You can monitor that interview in the Paused and Waiting Interviews list by looking for the type "Workflow".

Type	Description
Workflow	A running instance of a process created in the Process Builder.

SEE ALSO:

- Flow Properties
- Flow and Flow Version Fields

Flow Properties

A flow's properties consist of its name, description, interview label, and type. These properties drive the field values that appear on a flow or flow version's detail page. The properties of a flow and its flow versions are separate.

Tip: The properties for a given flow's versions automatically match the active version's properties by default. In other words, if you have three versions and you activate version 2, Salesforce updates the properties for versions 1 and 3 to match version 2. However, if you edit the properties for an inactive version, that version's properties are no longer automatically updated to match the active version.

From the Cloud Flow Designer, click  to update the properties for a flow or a flow version.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Property	Description
`Name`	The name for the flow or flow version. The name appears in the flow management page and flow detail page. It also appears in the run time user interface. You can edit the name for inactive flows and flow versions.
`Unique Name`	The unique name for the flow. The unique name is used to refer to this flow from other parts of Salesforce, such as in a URL or Visualforce page. A unique name is limited to underscores and alphanumeric characters. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. The unique name appears on the flow detail page. You can't edit the unique name after the flow has been saved.
`Description`	The description for the flow or flow version. The description appears in the flow management page and flow detail page. You can edit the description for inactive flows and flow versions.
`Type`	The type for the flow or flow version. The type appears in the flow management page and flow detail page. It determines which elements and resources are supported in the flow or flow version, as well as the ways that the flow can be implemented. For details, see Flow Types on page 583. If the type is Login Flow, you can't update the type after the flow has been saved.
`Interview Label`	The label for the flow's interviews. An *interview* is a running instance of a flow. This label appears in: • The Paused and Waiting Interviews list on the flow management page • The Paused Interviews component on the Home tab • The Paused Interviews item in Salesforce1 You can edit the interview label for inactive flows and flow versions. By default, the interview label contains the flow name and the `{!$Flow.CurrentDateTime}` system variable. Use a text template to reference multiple resources in the label. For example, ***`Flow Name`*** `- {!Account.Name} - {!$Flow.CurrentDateTime}.`

SEE ALSO:

Save a Flow

Flow and Flow Version Fields

Sample Flows

Sometimes showing is better than telling. Check out these sample flows to get a feel for how to work with advanced things like Wait elements and collection variables.

IN THIS SECTION:

Sample Flow That Populates a Collection Variable

Populate a collection variable by populating an sObject collection variable. Then individually assign the sObject collection variable's values to the collection variable.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

[Sample Flows That Wait for Events]
Configure a flow to wait for events in one of three ways.

[Sample Flow That Loops Through a Collection]
Transfer ownership of accounts from one user to another by using sObject variable collections and loops. The flow already has the required user IDs.

Sample Flow That Populates a Collection Variable

Populate a collection variable by populating an sObject collection variable. Then individually assign the sObject collection variable's values to the collection variable.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Scenario

In this scenario, you're designing a flow to send an email to every employee who lives in San Francisco.

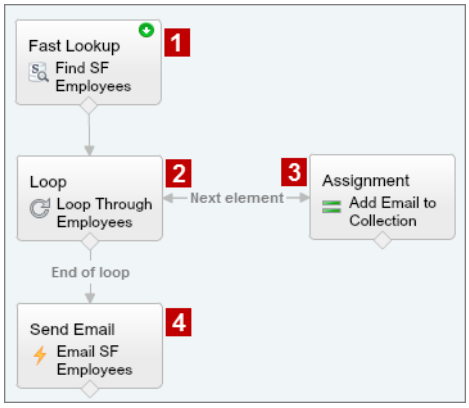
The Send Email element allows you to easily send emails from a flow. However, the Recipients parameter only accepts text variables and text collection variables. Since multiple users live in San Francisco, use a collection variable (rather than entering the email address for each individual user).

You can't use a Fast Lookup or Record Lookup to populate collection variables. First populate a User-based sObject collection variable with field values, including `Email`, from the employees who live in San Francisco. Then add those emails to the collection variable.

Once the collection variable is populated, you simply use the collection variable as the value for the Send Email element's `Email Addresses (collection)` parameter.

Example: This flow already contains these resources.

- A User-based sObject collection variable called `employeesInSF`
- A User-based sObject variable called `loopVariable`
- A Text-based collection variable called `emails_employeesInSF`



The example flow:

1. Finds all user records whose `City` is "San Francisco" and populates `employeesInSF` with those records' `Email`.

   Because `employeesInSF` is an sObject collection variable, use a Fast Lookup element to populate the variable.

2. Loops through the sObject collection variable so that it can look at each individual user record. The loop copies the values of each item in `employeesInSF` to `loopVariable`.
3. For each iteration, assigns the user's `Email` to a collection variable that has a Data Type of Text.
4. When the loop ends, the flow sends an email to the users whose emails are now stored in `emails_employeesInSF`.

SEE ALSO:

Flow Collection Variable Resource

Add Values to a Collection Variable

Sample Flows That Wait for Events

Configure a flow to wait for events in one of three ways.

IN THIS SECTION:

Sample Flow That Waits for Many Events

This flow waits for many events to occur, rather than just the first event. The base times for these events are field values, so this example uses relative time alarms.

Sample Flow That Waits for Only the First Event

This flow waits for the first of multiple events to occur before proceeding. The base times for these events are field values, so this example uses relative time alarms.

Sample Flow That Waits for a Single Event

This flow waits for a single event. The base time for the event in this example, which is an absolute alarm, is the `{!$Flow.CurrentDateTime}` system variable.

SEE ALSO:

Flow Wait Element

Sample Flow That Waits for Many Events

This flow waits for many events to occur, rather than just the first event. The base times for these events are field values, so this example uses relative time alarms.

You're designing a flow that reminds contract owners to follow up with their customers before the contract ends. Rather than sending just one reminder, however, the flow sends them regularly. This example shows how to use one Wait element to send a reminder two weeks before and then again one week before the contract ends. You could easily extend this flow to send reminders at more intervals, such as three days and one day before the contract ends.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

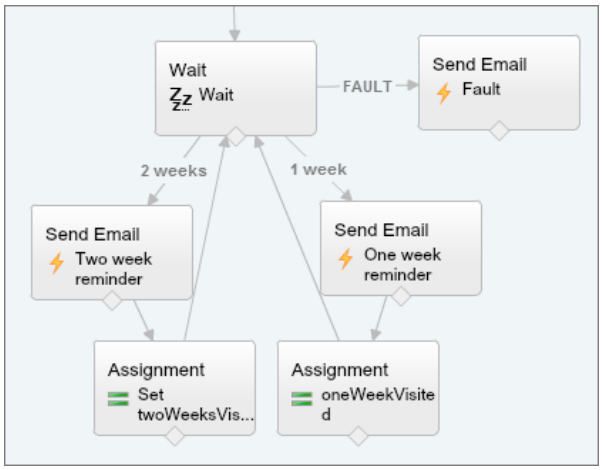
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example

This flow already contains these populated variables.

- `{!contract}` is an sObject variable that contains the contract's `Id` and `OwnerId`
- `{!oneWeekVisited}` is a Boolean variable whose default value is *`{!$GlobalConstant.False}`*
- `{!twoWeeksVisited}` is a Boolean variable whose default value is *`{!$GlobalConstant.False}`*

Before the flow executes the Wait element, it looks up and stores the contract's `Id` and `OwnerId`.

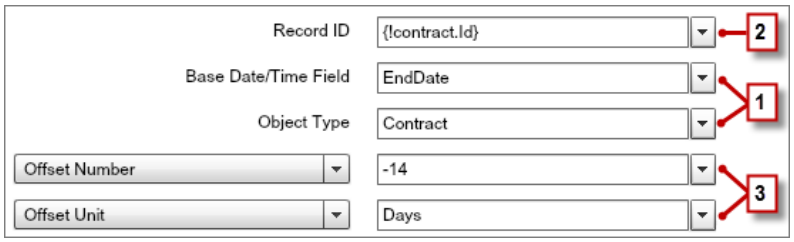


Because the flow sends the reminder emails both two weeks and a week before the contract's end date, the Wait element defines two relative alarm events.

Tip: Every alarm event consists of a base time and an offset. With relative time alarms, the flow needs three pieces of information to determine the base time: the object, the date/time field, and the specific record. The offset for relative time alarms works the same as it does for absolute time alarms. The flow needs to know the unit (either `Days` or `Hours`) and the number of those units. To wait for a number of days or hours before the base time, set `Offset Number` to a negative integer.

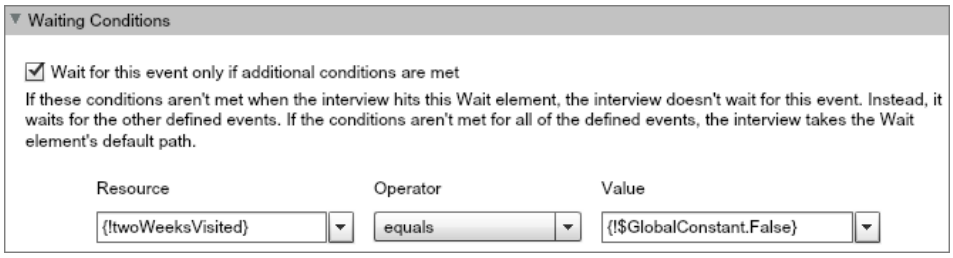
For both of these events, the offset is declared in `Days`, because weeks isn't an acceptable offset unit.

The base time for the first event ("2 Weeks") is the value of `Contract.EndDate` **(1)** on the record whose ID is stored in `{!contract.Id}` **(2)**. The offset is -14 days **(3)** to represent two weeks.



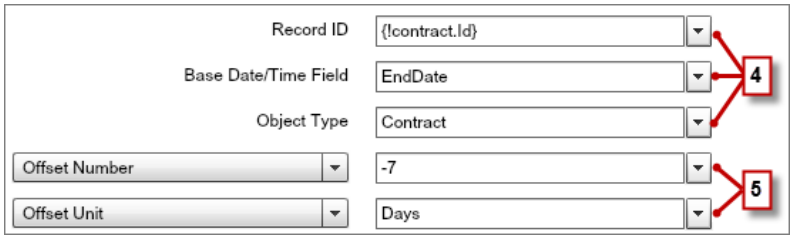
You want to use the same Wait element for every reminder, so after a flow interview sends one email reminder, it returns to the Wait element. But first, to ensure that the interview doesn't send the same email again and again, use *waiting conditions*. When an interview executes a Wait element, it first checks the waiting conditions for each event to determine whether to wait for those events. If an event has waiting conditions set and those conditions aren't met, the interview doesn't wait for that event.

For the first event, the interview checks whether the Boolean variable `{!twoWeekVisited}` is set to false. The variable's default value is set to `{!$GlobalConstant.False}`, so the flow waits for the event until the variable's value is changed.

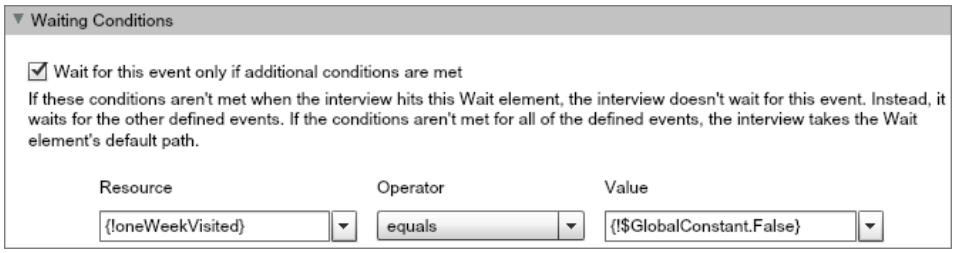


Indicate what the flow does when the "2 Weeks" event occurs by connecting the Wait element to other elements. Then, before you return the flow path to the Wait element, change the value of `{!twoWeeksVisited}` to `{!$GlobalConstant.True}`. You can do so with an Assignment element. If the value for `{!twoWeeksVisited}` isn't false when the Wait element is executed, the flow doesn't wait for the "2 Weeks" event to occur. Essentially, the interview checks whether the first event has occurred yet, since the variable is changed to true only in that event's path. If that event has occurred (and the variable isn't set to false), the interview knows not to wait for that event.

The second event ("1 Week") has the same base time as the first event **(4)**; the offset is -7 days **(5)** to represent a week.



For the second event, the flow checks whether the Boolean variable `{!oneWeekVisited}` is set to false. If it isn't, the flow doesn't wait for this event.



Like with the first event, use an Assignment element to change the value of `{!oneWeekVisited}` to `{!$GlobalConstant.True}` before the flow path returns to the Wait element. As long as `{!oneWeekVisited}` isn't false, the flow doesn't wait for the "1 Weeks" event to occur.

Tip: When a flow executes a Wait element and all the events have waiting conditions that aren't met, the flow executes the *default event path*. Because this flow is finished after it sends the final reminder, don't connect the default path to another element.

Just in case something goes wrong, set a fault path. In this example, the fault path sends an email that contains the fault message to the user who created the flow.

Sample Flow That Waits for Only the First Event

This flow waits for the first of multiple events to occur before proceeding. The base times for these events are field values, so this example uses relative time alarms.

You're designing a flow that reminds account owners to follow up with their customers a week before either the account renews or the contract ends. The flow sends a reminder email for whichever date occurs first.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

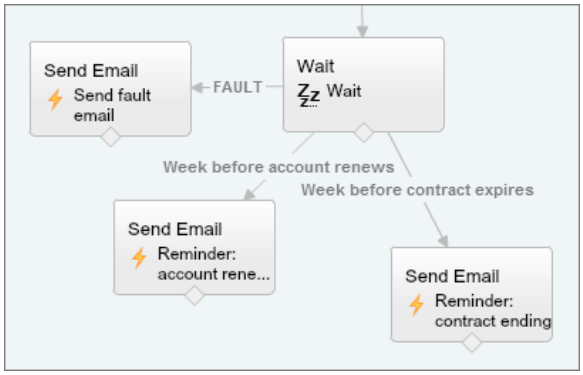
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example

This flow already contains these populated variables.

- `{!accountId}` contains the ID for the account
- `{!contractId}` contains the ID for the contract
- `{!accountOwner}` contains the ID for the account's owner
- `{!ownerEmail}` contains the account owner's email address

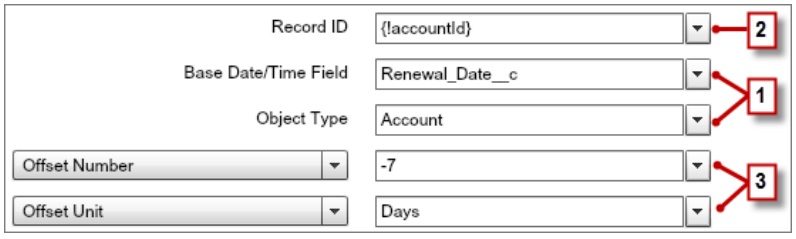
Before the flow executes the Wait element, it looks up and stores the contract's `ID`, its parent account's `ID` and `OwnerId`, and the account owner's `Email`.



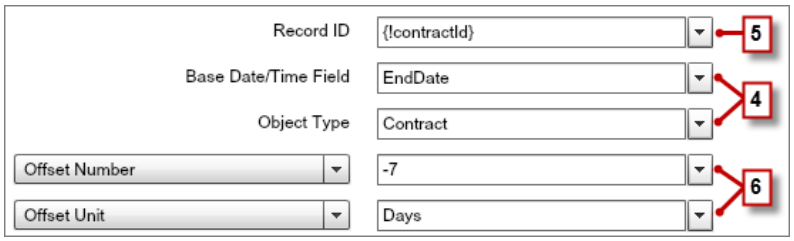
The Wait element defines two relative alarm events.

Tip: Every alarm event consists of a base time and an offset. With relative time alarms, the flow needs three pieces of information to determine the base time: the object, the date/time field, and the specific record. The offset for relative time alarms works the same as it does for absolute time alarms. The flow needs to know the unit (either *`Days`* or *`Hours`*) and the number of those units. For both of these events, the base time is offset by -7 days, because weeks isn't an acceptable offset unit.

The base time for the first event ("Week before account renews") is the value of `Account.Renewal_Date__c` **(1)** on the record whose ID is stored in `{!accountId}` **(2)**. The offset is -7 days **(3)**.



The base time for the second event ("Week before contract expires") is the value of `Contract.EndDate` **(4)** on the record whose ID is stored in `{!contractId}` **(5)**. The offset is -7 days **(6)**.



You only want to send one follow-up reminder and the flow always waits for both events, so neither of these events need waiting conditions. However, just in case something goes wrong, set a fault path. In this example, the fault path sends an email that contains the fault message to the user who created the flow.

SEE ALSO:

- Flow Wait Element
- Relative Time Alarms
- Flow Wait Element
- Relative Time Alarms
- What Are Waiting Conditions?

Sample Flow That Waits for a Single Event

This flow waits for a single event. The base time for the event in this example, which is an absolute alarm, is the `{!$Flow.CurrentDateTime}` system variable.

You're designing a flow that requests feedback from customers after a contract is activated, but you want to delay the email by a day.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

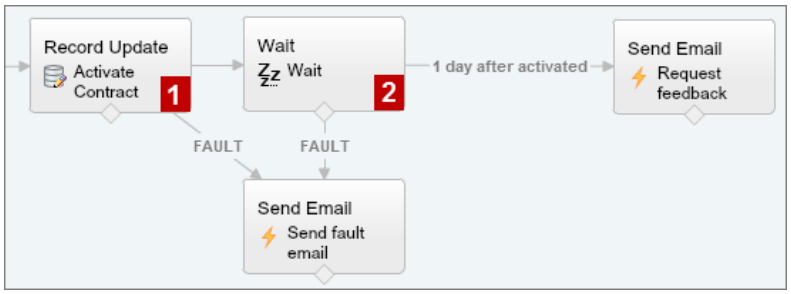
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example

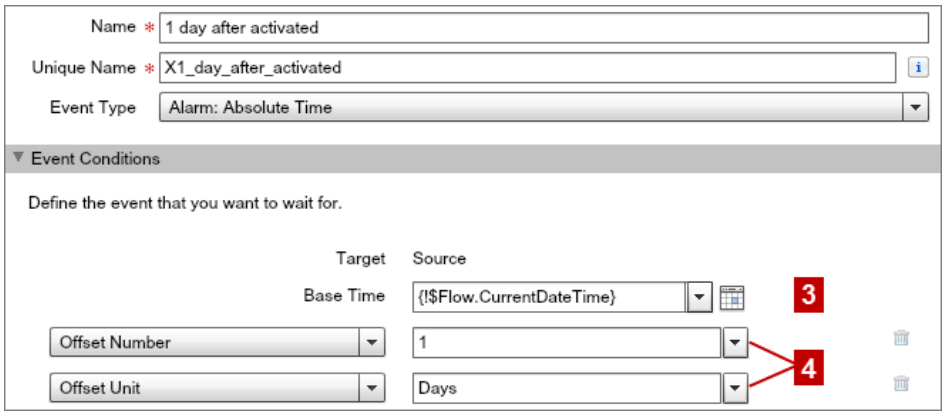
This flow already contains the following populated variables.

- `{!customerEmail}` contains the email address for the customer
- `{!creatorEmail}` contains the email address for the flow's creator

The flow activates a contract **(1)** and then waits **(2)**.



Within the Wait element, a single event is defined (1 day after activated). The flow sends the feedback request one day after the contract is activated, so use an absolute time alarm. The base time is the `{!$Flow.CurrentDateTime}` system variable **(3)**, and the offset is one day **(4)**.



Because there's only one event and you only want the feedback request to be sent once, don't set any waiting conditions for this event. However, just in case something goes wrong, don't forget to set a fault path. In this example, the fault path sends an email that contains the fault message to the user who created the flow.

SEE ALSO:

- Flow Wait Element
- Absolute Time Alarms

Sample Flow That Loops Through a Collection

Transfer ownership of accounts from one user to another by using sObject variable collections and loops. The flow already has the required user IDs.

First, create an Account-based sObject collection variable called `collAcctJSmith` and populate it with all account records that John Smith owns.

Then create a loop that iterates through the collection. For each item in the collection, the loop does the following:

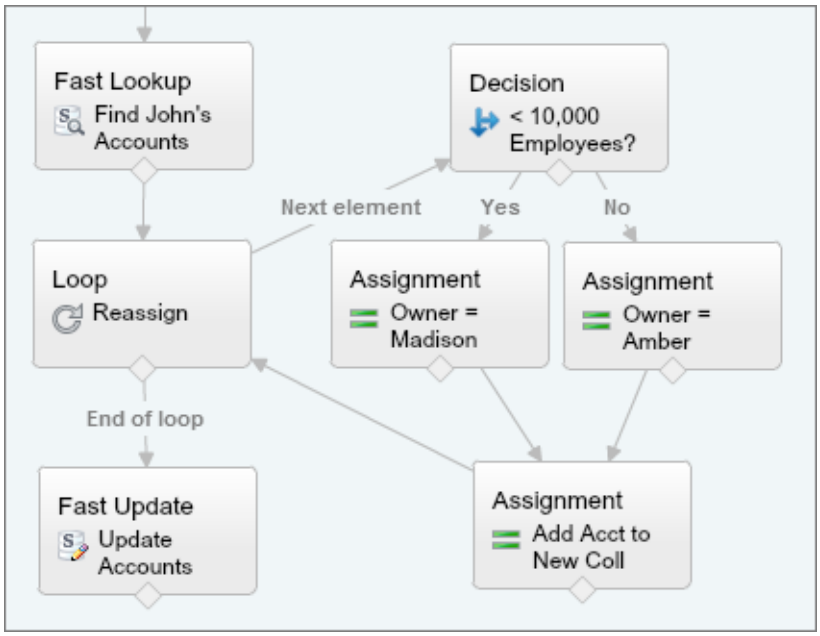
1. Assigns the collection item to the loop variable.
2. Evaluates whether the account has more than 10,000 employees.
3. If the account has more than 10,000 employees, assigns Madison's user ID to the `OwnerId` field in the loop variable.
4. If the account doesn't have more than 10,000 employees, assigns Amber's user ID to the `OwnerId` field in the loop variable.
5. Adds the loop variable's values as a new item in a second collection called `collReassignedAccts`.

Finally, create a Fast Update element to update the accounts in `collReassignedAccts` with the new `OwnerId` after the loop finishes iterating through the collection.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



This section of the flow uses a single query to look up the list of accounts and a single DML statement to update those accounts. If you created a similar flow by using Record Update elements, you would use:

- One Record Update element to find all accounts that John owns and have more than 10,000 employees (1 query). Then update those records' `OwnerId` to Madison's Id (1 DML statement).
- One Record Update element to find all accounts that John owns and don't have more than 10,000 employees (1 query). Then update those records' `OwnerId` to Amber's Id (1 DML statement).

Manage Your Flows

Use the flow detail page to do anything with your flow outside of designing it—such as activating a flow, testing it, or viewing its properties.

To visit a flow's detail page, from Setup, enter *`Flows`* in the `Quick Find` box, select **Flows**, and then click a flow name.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Flow and Flow Version Fields
View information about a flow and its versions on the flow detail page, like its name and URL.

Open and Modify a Flow
To modify a flow, open it in the Cloud Flow Designer.

Test a Flow
Test your flows before you activate them to make sure they're working as expected.

Activate or Deactivate a Flow Version
You can have several different versions of a single flow in Salesforce, but only one version of each flow can be active at a time. To activate or deactivate a version of a flow, go to that flow's detail page in Setup.

Delete a Paused or Waiting Flow Interview

If you no longer need to wait for a long-running flow interview to finish or for a user to resume a paused interview, delete the interview. For example, when you're updating or deleting the associated flow version.

Delete a Flow Version

To delete an active flow version, first deactivate it. If a flow has any paused or waiting interviews, it can't be deleted until those interviews are finished or deleted. Flows that have never been activated can be deleted immediately.

Let Users Pause Flows

Enable your users to pause a flow interview that they can't finish yet by customizing your organization's process automation settings. A *flow interview* is a running instance of a flow. For example, a customer service representative can pause a flow interview when the customer doesn't have all the necessary information.

SEE ALSO:

Visual Workflow

Limits for Visual Workflow

Considerations for Managing Flows

Flow and Flow Version Fields

View information about a flow and its versions on the flow detail page, like its name and URL.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Property	Description
`Active Version`	Identifies which version is active.
`Description`	The description for the flow or flow version
`Flow Name`	The name for the flow. It appears in the run time user interface.
`Name`	The name for the flow version. It becomes the `Flow Name` when this version is active.
`Namespace Prefix`	The flow's namespace prefix, if it was installed from a managed package. The Cloud Flow Designer can't open flows that are installed from managed packages.
`Type`	Determines which elements and resources are supported in the flow or flow version, as well as the ways that the flow can be distributed. For details, see Flow Types on page 583.
`Status`	Identifies whether the flow version is active or not.
`Unique Name`	Lets you refer to the flow from other parts of Salesforce, such as in Visualforce page.
`URL`	The relative URL that you can use to run the flow, such as from a custom button or Web tab.

Property	Description
`Version`	The number of the flow version.

SEE ALSO:

- Manage Your Flows
- Flow Properties

Open and Modify a Flow

To modify a flow, open it in the Cloud Flow Designer.

You can't save changes to an active flow version. You can, however, open an active version of a flow, modify it, and then save as a new version or a new flow.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.
2. Click the name of the flow.
3. Open the flow.
   - To open a specific version, click the **Open** link next to that version number.
   - To open the active version of the flow, click the **Open** button. If there isn't an active version, the latest version opens.

SEE ALSO:

- Considerations for Designing Flows
- Manage Your Flows
- Activate or Deactivate a Flow Version

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Test a Flow

Test your flows before you activate them to make sure they're working as expected.

Warning: Be careful when testing flows that contain delete elements. Even if the flow is inactive, it triggers the delete operation.

We recommend that you test all possible paths through the flow, so that you can find and fix any errors before activating the flow. For example, incomplete data in the flow can cause a data element (create, update, lookup, or delete) to fail at run time. Add a fault connector to a path that corrects the data and allows the flow to successfully finish.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.
2. Click the name of the flow you want to run.
3. Run the flow.
   - To run a specific version, click the **Run** link for that version.
   - To run the active version of the flow, click the **Run** button. If there isn't an active version, the latest version runs.
   - To run a flow version from the Cloud Flow Designer, open that version and then click **Run** from the button bar.

     Tip: If you recently modified the flow that you're testing, save it. Only the most recently saved changes are included when you run a flow.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To run an active or inactive flow from the flow detail page:

To run a flow from within the Cloud Flow Designer:

- "Manage Force.com Flow"

Once you're confident that your flow is working as expected, activate the version that you tested and then distribute the flow.

SEE ALSO:

- Activate or Deactivate a Flow Version
- Customize What Happens When a Flow Fails
- Considerations for Running Flows
- Manage Your Flows

Activate or Deactivate a Flow Version

You can have several different versions of a single flow in Salesforce, but only one version of each flow can be active at a time. To activate or deactivate a version of a flow, go to that flow's detail page in Setup.

When you activate a new version of a flow, the previously activated version (if one exists) is automatically deactivated. Any running flow interview continues to run using the version with which it was initiated.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.
2. Click the name of the flow.
3. Click **Activate** or **Deactivate** next to the relevant version of the flow.

SEE ALSO:

- Considerations for Managing Flows
- Manage Your Flows

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To activate or deactivate a flow:

- "Manage Force.com Flow"

Delete a Paused or Waiting Flow Interview

If you no longer need to wait for a long-running flow interview to finish or for a user to resume a paused interview, delete the interview. For example, when you're updating or deleting the associated flow version.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.

   If there are waiting interviews for any of your flows, the Paused and Waiting Interviews related list appears underneath the list of flows.

2. For each interview that you want to delete, click **Del**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

Delete a Flow Version

To delete an active flow version, first deactivate it. If a flow has any paused or waiting interviews, it can't be deleted until those interviews are finished or deleted. Flows that have never been activated can be deleted immediately.

1. From Setup, enter *Flows* in the Quick Find box, then select **Flows**.
2. Click the name of the flow.
3. To delete the flow completely, including all versions, click the **Delete** button.
4. To delete an individual version, click the **Del** link for that version.

SEE ALSO:

Considerations for Managing Flows

Manage Your Flows

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To delete a flow:

- "Manage Force.com Flow"

Let Users Pause Flows

Enable your users to pause a flow interview that they can't finish yet by customizing your organization's process automation settings. A *flow interview* is a running instance of a flow. For example, a customer service representative can pause a flow interview when the customer doesn't have all the necessary information.

1. From Setup, enter `Process Automation Settings` in the `Quick Find` box, then select **Process Automation Settings**.
2. Select `Let Users Pause Flows`.
3. Click **Save**.

Screens don't automatically display the Pause button once `Let Users Pause Flows` is enabled. If you want your users to be able to pause at a given screen, select "Show Pause button" when you configure that screen.

SEE ALSO:

Flow Screen Element: General Info

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit process automation settings:

- "Customize Application"

Distribute Your Flow

Once you've created and activated a flow version, distribute it to users so that they can run it. The right distribution method depends on the users that you want to distribute the flow to: internal users, external users, systems, or other organizations.

Tip: Users can't resume paused flows from Lightning Experience, so we recommend removing the Pause button from flows that are distributed in Lightning Experience.

Here are the ways that you can distribute your flows, based on the type of user that you're distributing to.

Internal users

- Distribute a flow URL directly or through a custom button, link, or Web tab.
- Embed the flow in a Lightning Page (Beta)
- Embed the flow in a Visualforce page and incorporate that page into Salesforce with a custom button, link, or Visualforce tab.

External users

Embed the flow in a Visualforce page, and incorporate that page into a Force.com site, Customer Portal, or Partner Portal.

Systems

Start a flow automatically by:

- Using the Apex `start()` method.
- Using the Invocable Action resource in the Force.com REST API.
- Adding a flow trigger workflow action to a workflow rule (Pilot).
- Adding a flow action to a process in the Process Builder.

Other organizations

Deploy a flow with change sets or a package.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

For details about distributing flows with each of these methods, see "Distribute Your Flow" in the *Visual Workflow Guide*.

SEE ALSO:

- Set Flow Variables with the Flow URL
- *Visualforce Developer Guide*: Render Flows with Visualforce
- Managing Force.com Site Visualforce Pages
- Create Web Tabs

Flow Runtime Experiences (Beta)

Depending on how the flow is distributed, your users see one of two UIs when they run a flow: *Classic runtime* or *Lightning runtime*. Like its name suggests, Lightning runtime looks and feels like Lightning Experience.

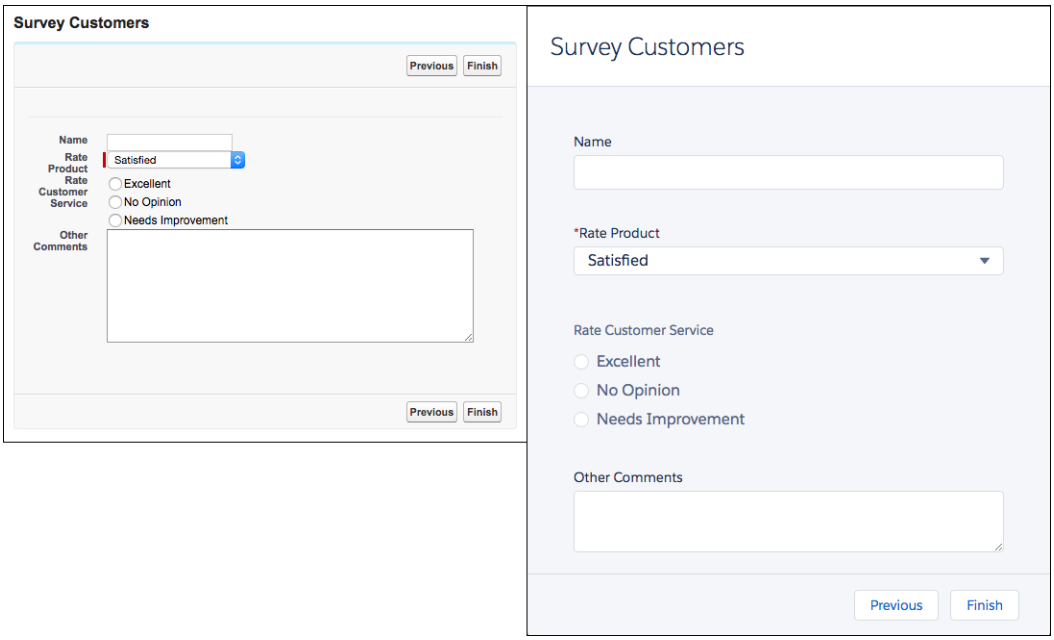
Note: This release contains a beta version of Flow Lightning Runtime, which means it's a high-quality feature with known limitations. Flow Lightning Runtime isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for the Flow Lightning Runtime in the IdeaExchange.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The following screenshot shows the same flow rendered in Classic runtime (left) and Lightning runtime (right).



Which Runtime Experience Will My Users See?

Flows run from a Visualforce page always use Classic runtime. Flows run from a Lightning page always use Lightning runtime. All other methods depend on whether Lightning runtime has been enabled in your org's Process Automation settings.

This table summarizes which runtime experience your users will see, based on how you distribute the flow.

Flow Distribution Method	When `Lightning Runtime for Flows` is	
	Not selected	Selected
Visualforce page	Classic runtime	Classic runtime
Custom button	Classic runtime	Lightning runtime
Custom link	Classic runtime	Lightning runtime
Web tab	Classic runtime	Lightning runtime
Direct link	Classic runtime	Lightning runtime
Lightning page	Lightning runtime	Lightning runtime

Do the Runtime Experiences Behave Differently?

For the most part, the only difference between the two runtime experiences is the look and feel. However, Lightning runtime doesn't support passing values to these types of variables from outside the flow.

- Picklist variables
- Multi-select picklist variables
- sObject variables
- Collection variables of any data type

SEE ALSO:

Choose Your Org's Runtime Experience for URL-Based Flows (Beta)

Considerations for Running Flows

Choose Your Org's Runtime Experience for URL-Based Flows (Beta)

Are you distributing a flow via a URL? That includes things like direct URLs and custom buttons, as well as links in Setup. Flip one switch, and all those flows will be upgraded to Lightning runtime.

Note: This release contains a beta version of Flow Lightning Runtime, which means it's a high-quality feature with known limitations. Flow Lightning Runtime isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for the Flow Lightning Runtime in the IdeaExchange.

We have two flavors of runtime experience for your flow users. *Classic runtime* looks more like a standard Visualforce page. *Lightning runtime* fits right in with Lightning Experience.

To render all URL-based flows in Lightning runtime:

1. From Setup, enter *`Process Automation Settings`* in the `Quick Find` box, then select **Process Automation Settings**.
2. Select **Enable Lightning Runtime for Flows**.
3. Save your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit process automation settings:

- "Customize Application"

When enabled, flows use Lightning runtime when they're run from:

- A direct link
- A custom button or link
- The **Run** button in the Cloud Flow Designer
- A **Run** link on the flow list page
- The **Run** button on a flow detail page

This setting also enables you to control whether a flow displays in one or two columns if you distribute the flow via a URL or via a Lightning Page.

SEE ALSO:

Flow Runtime Experiences (Beta)

Render Two-Column Screens from a Flow URL (Beta)

Render Two-Column Screens from a Flow URL (Beta)

When you distribute a flow using a URL, you can control whether to display the screens with one column or two columns. Two-column screens are supported only for orgs that use Lightning runtime.

Note: This release contains a beta version of Two-Column Flows, which means it's a high-quality feature with known limitations. Two-Column Flows isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for Two-Column Flows in the IdeaExchange.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Prerequisites

Enable Lightning runtime so that your flows will respect the specified layout.

1. From Setup, go to Process Automation Settings.
2. Select **Enable Lightning Runtime for Flows**.

Format

To display a flow's screens in two columns:

```
/flow/flowName?flowLayout=twoColumn
```

Examples

This example displays a "Case Management" flow in two columns.

```
/flow/Case_Management?flowLayout=twoColumn
```

This example displays a "User Info" flow in two columns and sets the varUserFirst and varUserLast variables (both of type Text) to the running user's FirstNameand LastName field values.

```
/flow/User_Info?varUserFirst={!$User.FirstName}&varUserLast={!$User.LastName}&flowLayout=twoColumn
```

SEE ALSO:

- Choose Your Org's Runtime Experience for URL-Based Flows (Beta)
- Considerations for Two-Column Flows (Beta)

Set Flow Variables with the Flow URL

When you distribute a flow using a URL, you can set variables within that flow using parameters in the URL.

You can't set the values for sObject variables and sObject collection variables using URL parameters. If the flow uses Lightning runtime, you also can't set the values for picklist variables, multi-select picklist variables, and collection variables of any data type. The variable must have its `Input/Output Type` set to allow input access.

These steps assume you're using a relative URL in your org to direct users to the flow.

1. From Setup, enter *`Flows`* in the `Quick Find` box, then select **Flows**.
2. Click the name of the flow.
3. Copy the URL for the flow.
4. At the end of the flow URL, append `?`***`name`***`=`***`value`***, where *`name`* is the unique name of the variable or collection variable in the flow that you want to set and *`value`* is the value you want to set it to. If you want to set multiple variable values, append `?`***`name1`***`=`***`value1`***`&`***`name2`***`=`***`value2`*** to the end of the flow URL. If you want to set the values for multiple items in the same collection variable, append `?`***`name`***`=`***`value1`***`&`***`name`***`=`***`value2`***.
   For example:

```
/flow/MyFlow?varNumber=100&varString=Hello
```

Here are the valid values for each flow variable and collection variable based on its data type.

Variable Type	Acceptable Values
Date	Merge field of type Date or `YYYY-MM-DD`
DateTime	Merge field of type Date/Time or `YYYY-MM-DDThh:mm:ssZ`
Text	Merge field of any type or a string
Number	Merge field of type Number or a numeric value
Currency	Merge field of type Number or a numeric value
Boolean	• Merge field of type Checkbox • True values: `true` or `1` • False values: `false` or `0`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view a flow's detail page:
- "View Setup and Configuration"

To create and edit custom buttons, links, and Web tabs:
- "Customize Application"

Note: When you distribute a flow, don't pass a currency field value from a Salesforce record into a flow Currency variable with a URL parameter. When a currency field is referenced through a merge field (such as `{!Account.AnnualRevenue}`), the value includes the unit of currency's symbol (for example, $). Flow variables of type Currency can accept only numeric values, so the flow fails at run time. Instead, pass the record's ID to a flow Text variable with a URL parameter. Then in the flow, use the ID to look up that record's value for the currency field.

Example: The following example is a flow URL that is used in a custom button on a case page layout. When a user clicks that button, the flow launches with the varID variable (of type Text) set to the case record's `CaseNumber` field value.

```
/flow/Case_Management?varID={!Case.CaseNumber}
```

The following example sets the varUserFirst and varUserLast variables (both of type Text) to the running user's `FirstName` and `LastName` field values.

```
/flow/User_Info?varUserFirst={!$User.FirstName}&varUserLast={!$User.LastName}
```

The following example is a flow URL that is used in a custom button on a contact page layout. When a user clicks that button, the flow launches and adds text values from the contact as items in the {!collNames} text collection variable.

```
/flow/Contact_Info?collNames={!Contact.FirstName}&collNames={!Contact.LastName}
```

SEE ALSO:

Set Flow Finish Behavior with a Flow URL
Distribute Your Flow
Troubleshoot Flow URLs

Set Flow Finish Behavior with a Flow URL

By default, when a flow interview that uses screens finishes, a new interview for that flow begins and the user is redirected to the first screen. If you want users to be redirected to another page within Salesforce when they click **Finish**, use the `retURL` parameter in the flow URL.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Format

To redirect users to a specific page in Salesforce after they click **Finish**:

```
/flow/flowName?retURL=page_name
```

where *page_name* is a relative URL (the part of the URL that comes after `https://yourInstance.salesforce.com/`).

Limitations

- You can't redirect flow users to a URL that's external to your Salesforce organization.
- You can't use a flow variable as the value for the retURL parameter. If you want to use a flow variable to redirect a user, such as to a specific record, distribute the flow by using Visualforce.
- retURL can cause nested top and side navigation bars to render on the destination page.

Examples

The following flow URL redirects the user to the home tab for cases (`https://yourInstance.salesforce.com/500/o`).

```
/flow/Case_Management?retURL=500/o
```

The following flow URL sets the varUserFirst and varUserLast variables (both of type Text) to the running user's `FirstName` and `LastName` field values. When the flow interview finishes, the user is redirected to `https://yourInstance.salesforce.com/home/home.jsp`.

```
/flow/User_Info?varUserFirst={!$User.FirstName}
   &varUserLast={!$User.LastName}&retURL=home/home.jsp
```

SEE ALSO:

- Distribute Your Flow
- Troubleshoot Flow URLs
- Set Flow Variables with the Flow URL

Flows in Change Sets and Packages

Flows created in the Cloud Flow Designer can be included in change sets and packages. The recipient organization of either the change set or package must have Visual Workflow enabled.

IN THIS SECTION:

Considerations for Deploying Flows with Change Sets

Before you use change sets to deploy a flow, understand the limits and unexpected behaviors that are related to component dependencies, deployment, and flow triggers.

Considerations for Deploying Flows with Packages

Flows can be included in both managed and unmanaged packages. Before you deploy one, understand the limitations and behaviors of packages that contain flows.

SEE ALSO:

- Considerations for Installed Flows
- Limits and Considerations for Visual Workflow
- Distribute Your Flow

Considerations for Deploying Flows with Change Sets

Before you use change sets to deploy a flow, understand the limits and unexpected behaviors that are related to component dependencies, deployment, and flow triggers.

Component Dependencies

- If you plan to deploy a flow with change sets, consider limitations in migration support. Make sure your flows reference only fields and components that are available in change sets.
- When you view the dependent components for the change set, the Component Dependencies page lists the dependencies for *all* versions of the flow. Add all interdependent components for the relevant flow version to the outbound change set.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- If a component is referenced by the following flow elements, the Component Dependencies page doesn't display that component. To deploy the flow successfully, manually add those referenced components to the change set.
  - Apex
  - Email Alerts
  - Post to Chatter
  - Quick Actions
  - Send Email
  - Submit for Approval

  For example, if you use an email alert, manually add the email template that is used by that email alert.

Deployment

- You can include only one version of a flow in a change set.
- An active flow in a change set is deployed to its destination as inactive. Activate the flow manually after deployment.
- If the flow has no active version when you upload the outbound change set, the latest inactive version is used.
- Deploying or redeploying a flow with change sets creates a version of the flow in the destination organization.

Flow Triggers

Flow triggers aren't available in change sets.

The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Considerations for Deploying Flows with Packages

Flows can be included in both managed and unmanaged packages. Before you deploy one, understand the limitations and behaviors of packages that contain flows.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Component Dependencies

- If you plan to deploy a flow with packages, consider limitations in migration support. Make sure your flows reference only packageable components and fields.
- Referential integrity works the same for flows as it does for other packaged elements.
- If any of the following elements are used in a flow, packageable components that they reference aren't included in the package automatically. To deploy the package successfully, manually add those referenced components to the package.
  - Apex
  - Email Alerts
  - Post to Chatter
  - Quick Actions
  - Send Email
  - Submit for Approval

  For example, if you use an email alert, manually add the email template that is used by that email alert.

Flow Status

You can package only active flows. The active version of the flow is determined when you upload a package version. If none of the flow's versions are active, the upload fails.

Updating Packages

- To update a managed package with a different flow version, activate that version and upload the package again. You don't need to add the newly activated version to the package. However, if you activate a flow version by mistake and upload the package, you'll distribute that flow version to everyone. Be sure to verify which version you really want to upload.
- You can't include flows in package patches.

Other Limitations

- If you register your namespace after you referenced a flow in a Visualforce page or Apex code, don't forget to add the namespace to the flow name. Otherwise, the package will fail to install.
- If someone installs a flow from a managed package, error emails for that flow's interviews don't include any details about the individual flow elements. The email is sent to the user who installed the flow.
- Flow triggers aren't packageable.

  The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

- In a development organization, you can't delete a flow or flow version after you upload it to a released or beta managed package.

SEE ALSO:

Considerations for Installed Flows

Why Did My Flow Interview Fail?

To troubleshoot a failed flow interview, use the flow fault email. You can also set up temporary Screen or Send Email elements to identify the problem.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Emails About Flow Errors

Every time a flow interview fails, the admin who created the associated flow gets an email. The email includes the error message from the failure and details about every flow element that the interview executed.

Limitations of Emails About Flow Errors (Beta)

The email about errors in flow interviews has some limitations for Screen, Lookup, Create, and Subflow elements—as well as some general limitations.

Add Temporary Elements to a Flow

Add Screen or Send Email elements to the flow so you can check what the resources' values are at any given time. Once you've solved the problem, delete temporary Screen elements.

Troubleshoot Flow URLs

If you're distributing a flow and the custom button, custom link, or a direct flow URL isn't working as expected, verify the referenced flow. In addition, verify its variables if you're passing values into a flow from the URL.

Emails About Flow Errors

Every time a flow interview fails, the admin who created the associated flow gets an email. The email includes the error message from the failure and details about every flow element that the interview executed.

Note: This release contains a beta version of the flow error email that is production quality but has known limitations. You can provide feedback and suggestions for the flow error email on the IdeaExchange.

If the interview failed at multiple elements, the admin receives multiple emails, and the final email includes an error message for each failure. If a flow uses fault connectors, its interviews can fail at multiple elements.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Example:

```
An error occurred at element Apex_Plug_in_1.
List index out of bounds: 0.

An error occurred at element Fast_Delete_1.
DELETE --- There is nothing in Salesforce matching your delete criteria.

An error occurred at element Email_Alert_1.
Missing required input parameter: SObjectRowId.
```

SEE ALSO:

Limitations of Emails About Flow Errors (Beta)

Customize What Happens When a Flow Fails

Why Did My Flow Interview Fail?

Limitations of Emails About Flow Errors (Beta)

The email about errors in flow interviews has some limitations for Screen, Lookup, Create, and Subflow elements—as well as some general limitations.

Note: This release contains a beta version of the flow error email that is production quality but has known limitations. You can provide feedback and suggestions for the flow error email on the IdeaExchange.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

General

- If the user who started the flow doesn't have a first name, `null` replaces the user's first name in the "How the Interview Started" section.
- Variable assignments display in this pattern: `{!variable} (prior value) = field/variable (new value)`. If the variable had no prior value, the parentheses display as empty. For example: `{!varStatus} () = Status (Delivered)`
- If you install a flow from a managed package, error emails for that flow's interviews don't include any details about the individual flow elements. The email is sent to the user who installed the flow.

Screen elements

Password fields display in plain text, just like if you reference a password field in a Display Text field.

Lookup elements

The email displays lookup elements as "query" elements. Record Lookup displays as Record Query, and Fast Lookup displays as Fast Query.

Subflow elements

- The merge field annotation (`{!variable}` as opposed to just `variable`) is missing for variables in a referenced flow. For example, when an interview enters a subflow and gives details about the inputs, the subflow's variable is `subVariable` instead of `{!subVariable}`.
- If the error occurs in a referenced flow, the email gets sent to the author of the master flow, but the subject references the name of the referenced flow.
- If you see multiple "Entered flow *`ReferencedFlowName`* version *`ReferencedFlowVersion`*" messages with no "Exited *`ReferencedFlowName`* version *`ReferencedFlowVersion`*" messages in between them, the flow user navigated backwards. To prevent this scenario, adjust the navigation options in the first screen of the referenced flow so that the user can't click **Previous**.

SEE ALSO:

Emails About Flow Errors

Why Did My Flow Interview Fail?

Add Temporary Elements to a Flow

Add Screen or Send Email elements to the flow so you can check what the resources' values are at any given time. Once you've solved the problem, delete temporary Screen elements.

1. Create a single text template that contains the values of all resources and the fault message.
2. For each fault path, use the text template to configure one of the following elements.
   - A Screen element that uses the text template in a Display Text Field. (Only if the flow's type supports Screen elements.)
   - A Send Email element that uses the text template as the body and your email as the recipient.
   - A Post to Chatter element that uses the text template as the message. Consider creating a Chatter group specifically for flow errors.
3. Test the flow.

Example: Here's a text template for the Calculate Discounts on Opportunities flow in the Cloud Flow Designer Workbook.

```
RESOURCE VALUES for "Calculate Discounts on Opportunities"
opptyID: {!opptyID}
AccountID: {!AccountID}
AccountRevenue: {!AccountRevenue}
Full_Discount outcome: {!Full_Discount}
Partial_Discount outcome: {!Partial_Discount}
Discount: {!Discount}

ERROR
{!$Flow.FaultMessage}
```

After each element in the flow, add a temporary Post to Chatter element. Each Post to Chatter element is configured to use:

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

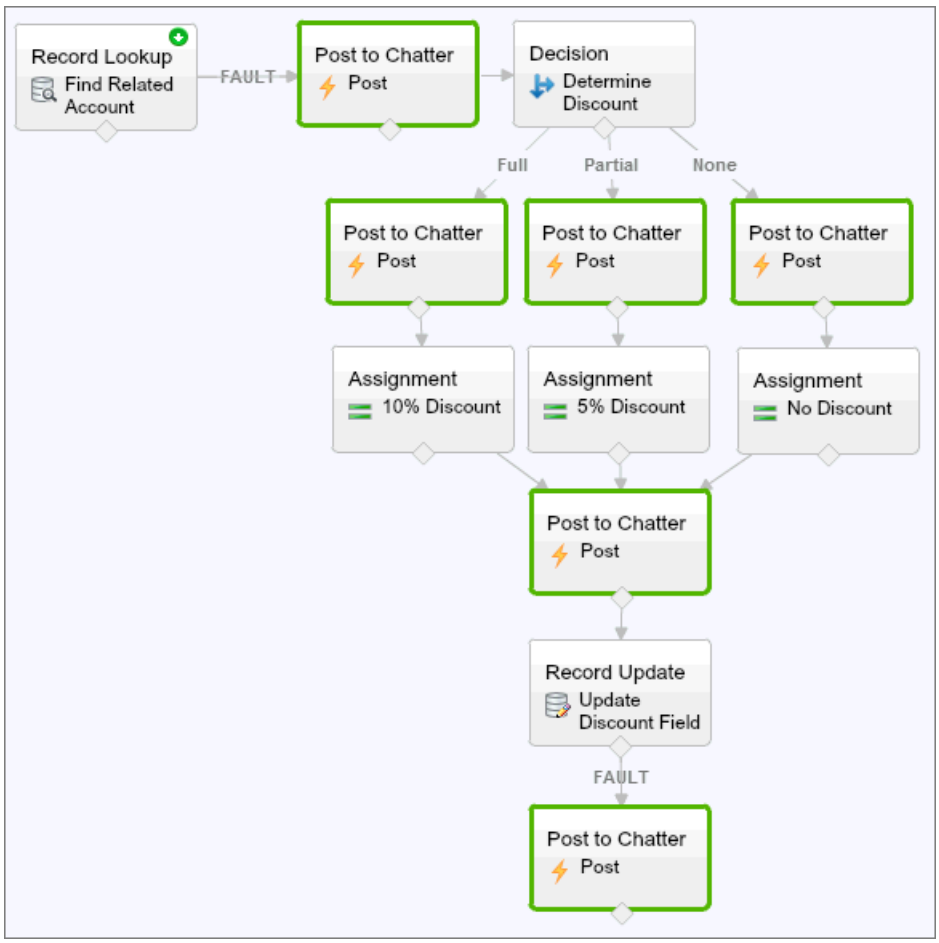
USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- "Manage Force.com Flow"

- The text template as the post's message
- The "Flow Troubleshooting" Chatter group as the post's target

Configure the Record Lookup and Record Update elements' fault connecters so that they route to the Post to Chatter elements.



This way, the flow posts to the Chatter group after the Record Lookup, Decision, Assignment, and Record Update elements are executed. Each post provides insight into the values of each resource throughout the flow. If the flow fails, the error is included in the Chatter posts.

After you identify and fix the problem with the flow, remove the temporary elements.

SEE ALSO:

Why Did My Flow Interview Fail?

Troubleshoot Flow URLs

If you're distributing a flow and the custom button, custom link, or a direct flow URL isn't working as expected, verify the referenced flow. In addition, verify its variables if you're passing values into a flow from the URL.

To make sure that the URL can find the right flow, verify that:

- The flow that the URL references hasn't been deleted or deactivated.
- The flow name is spelled and capitalized correctly. It must be an exact, case-sensitive match to the flow's `Unique Name`.

If your flow URL references a specific flow version, verify that the version hasn't been deleted or deactivated.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

If you're using the URL to pass values into the flow and the URL can't access the variable, the parameter that references the variable is ignored.

To make sure that the URL can find the right flow variable, verify that each variable you're trying to pass values into:

- Is spelled and capitalized correctly. It must be an exact, case-sensitive match to the variable.
- Allows input access. The variable's `Input/Output Type` must be set to "Input Only" or "Input and Output."
- Hasn't been renamed in the flow.
- Hasn't been removed from the flow.
- Isn't an sObject variable or an sObject collection variable.

In addition, make sure the value that you're trying to pass into the variable is compatible with the variable's data type and is correctly formatted.

SEE ALSO:

Set Flow Finish Behavior with a Flow URL

Set Flow Variables with the Flow URL

Why Did My Flow Interview Fail?

Flow Interviews

A *flow interview* is a running instance of a flow. A *flow* is an application built by your administrator that asks you for inputs and does something in Salesforce based on those inputs.

For example, a flow could provide a call script for customer support calls and use the information you provide to create a case. What the flow does with the information you provide is entirely up to your administrator.

When you run a flow interview, whether through a link, button, or tab, you're running a single instance of a flow. If the terminology is confusing, consider the difference between a record and an object. You create an account *record*, which is a single instance of the Account *object* that your administrator customized.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: Keep these tips in mind when you run a flow.

- Don't use your browser's Back or Forward buttons to navigate through a flow. Doing so can result in inconsistent data between the flow and Salesforce.
- A single flow can have up to 50 different versions. When you run a flow, you see the active version, but your admin could have a more recent version.

IN THIS SECTION:

Pause a Flow Interview

If your administrator has configured a given flow to do so, you can pause its interviews. Pausing is useful, for example, when a call with a customer drops or the customer can't find their account number and plans to call you back.

Resume a Flow Interview

If you paused a flow interview, resume it after you have all the necessary information.

Delete a Flow Interview

If you paused a flow interview and don't plan to resume it, delete it. By removing unnecessary interviews, you make sure that your pending list includes only interviews that you still plan to act on.

SEE ALSO:

Visual Workflow

Pause a Flow Interview

If your administrator has configured a given flow to do so, you can pause its interviews. Pausing is useful, for example, when a call with a customer drops or the customer can't find their account number and plans to call you back.

1. In an open flow interview, click **Pause**.
2. Explain why you had to pause the flow.

   This step is optional, but it helps differentiate between the different flow interviews that you've paused, especially if you paused multiple interviews of the same flow. This explanation can be up to 255 characters long.
3. Click **OK**.

The flow interview is saved until you resume or delete it later. Any valid values that you entered before you paused are saved with the interview, so you don't have to reenter that information when you resume.

SEE ALSO:

Flow Interviews

Resume a Flow Interview

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To pause a flow interview:

- "Run Flows"

  OR

  `Force.com Flow User` field enabled on the user detail page

Resume a Flow Interview

If you paused a flow interview, resume it after you have all the necessary information.

Note: If you pause an interview and the associated flow is updated before you resume, the resumed interview doesn't use the updated flow. Instead, it uses the version of the flow that was active when you paused.

1. Go to a place that lists all of your paused flow interviews.

   Depending on your administrator's configurations, you can find all the flow interviews that you've paused in one of these places.

   - "Paused Flow Interviews" on the Home tab in the full Salesforce site
   - "Paused Flow Interviews" in the left navigation in Salesforce1

   If you paused a flow interview and can't find the place where you're supposed to resume the interviews from, contact your administrator.

2. Resume the appropriate flow interview.

   If you entered values before you paused, all valid values are restored to those fields when you resume. If you don't see something that you entered before you paused, that value was invalid and the flow didn't save it.

   If you enter "Acme, Inc." for a field that only accepts numbers and then pause, that field is blank when you resume the flow interview.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To resume a flow interview:

- "Run Flows"

  OR

  `Force.com Flow User` field enabled on the user detail page

Warning: Once you resume a flow interview, that interview is removed from your Paused Flow Interviews list. If you resume an interview and then change your mind, click **Pause**. If you close the interview before pausing, you can't resume the interview later.

SEE ALSO:

Flow Interviews

Pause a Flow Interview

Delete a Flow Interview

Delete a Flow Interview

If you paused a flow interview and don't plan to resume it, delete it. By removing unnecessary interviews, you make sure that your pending list includes only interviews that you still plan to act on.

1. Go to a place that lists all of your paused flow interviews.

   Depending on your administrator's configurations, you can find all the flow interviews that you've paused in one of these places.

   - "Paused Flow Interviews" on the Home tab in the full Salesforce site
   - "Paused Flow Interviews" in the left navigation in Salesforce1

   If you paused a flow interview and can't find the place where you're supposed to resume the interviews from, contact your administrator.

2. Delete the flow interview that's unnecessary.

SEE ALSO:

Flow Interviews

Resume a Flow Interview

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To resume a flow interview:

- "Run Flows"

  OR

  `Force.com Flow User` field enabled on the user detail page

Visual Workflow Terminology

The following terminology is used for Visual Workflow in Salesforce:

Cloud Flow Designer
Cloud-based application that lets administrators create a flow for use in Salesforce.

Connector
Connectors determine the available paths that a flow can take at run time.

Element
Each *element* represents an action that the flow can execute. Examples of such actions include reading or writing Salesforce data, displaying information and collecting data from flow users, executing business logic, or manipulating data.

Flow
A *flow* is an application that can execute logic, interact with the Salesforce database, call Apex classes, and collect data from users. You can build flows by using the Cloud Flow Designer.

Flow Interview
A *flow interview* is a running instance of a flow.

Master Flow
A master flow is a flow that contains a subflow element. The term "master" is used to distinguish it from the flow that is referenced and called by the subflow element.

Resource
Each *resource* represents a value that you can reference throughout the flow.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Subflow

A subflow element references another flow, which it calls at run time. The flow that contains the subflow element is referred to as the master flow.

SEE ALSO:

Cloud Flow Designer

Visual Workflow

Automated Actions

An automated action is a reusable component that performs some sort of action behind the scenes—like updating a field or sending an email. Once you create an automated action, add it to a process, milestone, or other automated process.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Action Type	Supported In...				
	Workflow Rule	Process Builder	Flow	Approval Process	Entitlement Process
Email Alert	✅	✅	✅	✅	✅
Field Update	✅			✅	✅
Flow Trigger (Pilot)	✅ (immediate only)				
Outbound Message	✅			✅	✅
Task	✅			✅	✅

IN THIS SECTION:

Considerations for Automated Actions

Before you start working with automated actions, familiarize yourself with relevant limits and special behaviors.

Task Actions

Task actions determine the details of an assignment given to a specified user by an automated process. You can associate task actions with workflow rules, approval processes, or entitlement processes.

From Setup, enter *Tasks* in the Quick Find box, and select **Tasks**. Then use these settings to configure your task.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Object	Select an object for your task. Remember, tasks can only be associated with workflow rules or approval processes for the same object type. Choose an object that tracks activities. Note: Tasks are not available for article types.
Assigned to	Select an assignee for your task. An assignee can be in the form of a user, role, record owner, record creator, opportunity team role, or account team role, depending on the type of record you chose. Note: • If the assignee of a task is set to the record owner and the owner of a lead or case is a queue, the task is assigned to the person who triggered the rule. • If the subject was translated, Salesforce creates the task in the assignee's language. If you checked Notify Assignee for the task, Salesforce sends the email in the assignee's language. However, if only the comments field was translated, Salesforce creates the task and sends the email in the language of the user who triggered the task. • If a custom object has a master-detail relationship with a standard object, the owner of the custom object record is automatically set to the owner of the master standard object record. For example, if a custom object called “Expenses” has a master-detail relationship with Accounts, the owner of each expense record is the owner of the account for the expense. • If the assignee of a workflow task is a role and more than one user belongs to that role, the record owner becomes the task assignee, regardless of their role. We recommend that you not assign tasks to roles with multiple users. Assigning tasks to roles with one user allows you to easily change the user in that role without modifying the workflow rule. If the assignee of a workflow task is a role and that role is empty, the record owner becomes the task assignee, regardless of their role. • When a lead is converted by someone who isn't the lead owner, all workflow tasks associated with the lead that are assigned to that user, except email alerts, are reassigned to the lead owner. Workflow tasks assigned to users other than the lead owner and lead converter aren't changed.
Subject	Enter a subject for the task. Distinguish automated tasks from user-created ones by starting the subject with a specific notation, such as adding `(Automated)` at the end.
Unique Name	Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have actions of the same type with the same unique name, provided they are defined for different objects. The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Due Date	Choose a due date for the task. Due dates appear in the time zone of the assignee. Configuring a task’s **Due Date** to "Rule Trigger Date" sets time triggers and workflow task due dates based on the date that the workflow time trigger’s action is executed. For example, if the task due date is "Rule Trigger Date plus 10 days" and the time trigger is executed on January 1, Salesforce sets the task due date to January 11.
Status	Choose a status for the task.
Priority	Choose a priority for the task.

Field	Description
Comments	Enter comments for the task.

You may notice that all your tasks include a **Created By** field. For tasks, this field contains the name of the person who saved the record that triggered the rule to assign the task.

Tasks don't trigger task-based workflow rules if they're created automatically, such as by clicking the **Send An Email** button or by using the Email to Salesforce BCC address field.

SEE ALSO:

Associate Actions with Workflow Rules or Approval Processes

Email Alert Actions

Email alerts are emails generated by an automated process and sent to designated recipients. These actions consist of the standard text and list of recipients for an email. You can associate email alerts with processes, flows, workflow rules, approval processes, or entitlement processes. They're also available through the Invocable Actions REST API endpoint.

From Setup, enter *Email Alerts* in the Quick Find box, and select **Email Alerts**. Then use these settings to configure your email alert.

Tip: Create a standardized letterhead to use for all email templates you use for workflow alerts.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Description	Enter a description.
Unique Name	Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have actions of the same type with the same unique name, provided they are defined in different objects.The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Object	If available, choose an object for this email alert. Salesforce uses this object when generating merge field values for email templates with workflow rules and approval processes. Also, you can define the recipients of this email alert using contact and user lookup fields that are relevant to that object. For example, if you select Contract, you can define the contract signer as a recipient. The object is read-only if the new email alert is associated with an approval process or workflow rule for a particular object.

Field	Description
Email Template	Choose an email template. Insert merge fields to reference specific information based on the record that triggers the email alert. For example, insert a link to the opportunity. Except for `{!ApprovalRequest.Comments}`, approval merge fields named `{!ApprovalRequest.field_name}` in email templates return values only in approval assignment emails and email alerts for approval processes. When used in other emails—including email alerts for workflow rules—the approval merge fields return null. The `{!ApprovalRequest.Comments}` merge field returns only the most recently entered comment in emails for an approval step that requires unanimous approval from multiple approvers. If available, select **Protected Component** to mark the alert as protected if it is part of a Managed - Released package.
Recipient Type	Select who receives this email alert.
Recipients	Select who receives this email alert in the **Available Recipients** list and click **Add**. If you change the object after selecting recipients, Salesforce clears the **Selected Recipients** list. If your email recipient is a role and that role contains multiple people, Salesforce emails each person in that role. If your email recipient is a record owner and the owner of the record is a queue, the queue email receives the email alert. If the queue is set up so that email is sent to all members, queue members are notified as well. If no queue email is specified, only queue members are notified.
Additional Emails	Enter up to five additional email addresses for recipients who aren't Salesforce users, leads, or contacts.
From Email Address	Either the default workflow user or a previously configured and verified organization-wide address. This field lets you use a standard email address for your organization (such as support@company.com) instead of the default **From** field, which is the email address of the person who updates the record. If you select **Make this the default From email address for this object's email alerts**, this email address overrides the **From Email Address** for all email alerts associated with that object. You can still customize individual email alerts to use a different **From Email Address**. The **From Email Address** in a workflow email alert changes to the current user when the email alert is installed by using a managed or unmanaged package. The **From Email Address** doesn't change when using other types of deployment such as the Metadata API or change sets.

Note: The daily limit for emails sent through email alerts is 1,000 per standard Salesforce license per org—except for free Developer Edition and trial orgs, where the daily workflow email limit is 15. The overall org limit is 2,000,000. This limit applies to emails sent through email alerts in workflow rules, approval processes, flows, processes, or the REST API.

SEE ALSO:

Available Recipient Types for Email Alerts

Available Recipient Types for Email Alerts

When you configure an email alert, you identify who receives the email. The available options vary based on your org settings and the object you selected.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Recipient Type	Description
Account Owner	The user listed as the owner of the account itself or the account associated with the record. This option works only for email alerts on accounts, opportunities, cases, contacts, contracts, and any custom object that is a child of the account object. The associated account must also specify an account owner. If you select another object type or the associated account doesn't have an account owner, Salesforce sends the email alert to the record owner instead.
Account Team	All users assigned to a particular account team role. The Account Team option is always available. However, emails are sent only when the rule is associated with the account object or its immediate child objects.
Case Team	All users assigned to a particular case team role.
Creator	The user who created the record.
Customer Portal User	All users associated with a Customer Portal.
Email Field	An email address field on the selected object, such as the Email field on lead records or custom email fields. Note: When creating email alerts for campaign members, Email Field refers to the email field on the lead or contact that the campaign member is based on.
Opportunity Team	All users assigned to a particular opportunity team role. This option appears only when team selling is enabled. The Opportunity Team option works only for email alerts configured for opportunities. It doesn't work for email alerts configured for child objects of opportunities.
Owner	The record owner.
Partner User	All users associated with a partner portal.
Portal Role	All users assigned to a particular portal role.
Portal Role and Subordinates	All users assigned to a particular portal role, plus all users in roles below that role.
Public Groups	The users in a particular public group.

Recipient Type	Description
Related Contact	An associated contact on the record. For example, you can select the Customer Signed By field for contracts that contain the name of the contract signer.
Related Lead or Contact Owner	A campaign member's lead or contact owner.
Related User	An associated user on the record. For example, contract records have an Activated By field that contains the name of the user that activated the contract.
Role	All users assigned a particular role.
Role and Internal Subordinates	All users assigned a particular role, plus all users in roles below that role, excluding partner portal and Customer Portal users.
Role and Subordinates	All users assigned a particular role, plus all users in roles below that role.
User	A particular user.

Field Update Actions

Field update actions let you automatically update a field value. You can associate field updates with workflow rules, approval processes, or entitlement processes.

From Setup, enter *Field Updates* in the Update box, and select **Field Updates**. Then use these settings to configure your field update.

Before you begin, check the type of the field you want to update. Read-only fields like formula or auto-number fields are not available for field updates.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Name	Enter a name for this field update.
Unique Name	Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have field updates of the same type with the same unique name, provided they are defined in different objects.The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Description	Enter a description for the field update.
Object	Select the object whose field you want to update.
Field to Update	Select the field to update. Fields are shown only for the object that you selected. You can select a field on a related object in a master-detail relationship. You can use field updates on encrypted custom fields, but the encrypted field isn't available in the formula editor.

Field	Description
	Tip: Avoid associating more than one field update with a rule or approval process that applies different values to the same field.
Re-evaluate Workflow Rules After Field Change	Select if you want workflow rules on this object to be re-evaluated after the field value is updated. If you select this option, Salesforce re-evaluates all workflow rules on the object if the field update results in a change to the value of the field, triggering any workflow rules whose criteria are met. For more information, see Field Updates That Re-evaluate Workflow Rules on page 630.
Specify New Field Value	The value that the field should be updated with. The available options depend on the type of field you are updating. For more information, see Value Options for Field Update Actions on page 620 .

SEE ALSO:

Associate Actions with Workflow Rules or Approval Processes

Cross-Object Field Updates

Considerations for Field Update Actions

Value Options for Field Update Actions

When you create a field update action, specify what the new value of the field should be.

Available field update options depend on the type of field you're updating.

- Choose **A specific value**, and enter the value in the space provided.
- Choose **A blank value (null)** if you want Salesforce to remove any existing value and leave the field blank. This option isn't available for required fields, checkboxes, and some other types of fields.
- For record owners, choose the user to whom the record should be assigned. For case, lead, and custom object records, you can also choose a queue for this field. Select `Notify Assignee` to send an email to the new record owner (option unavailable when user control over task assignment notifications is enabled).
- For checkboxes, choose `True` to select the checkbox and `False` to deselect it.
- For picklists, select a specific value from the drop-down list, or select the value above or below the current value based on the sorting specified in the picklist definition. If you sort values alphabetically, the values above or below may be different for users in other languages.
- Choose **Use a formula to set the new value** to calculate the value based on an expression, merge fields, or other values. For more information about using formulas in Salesforce, see Formulas on page 229.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Outbound Message Actions

An outbound message sends information to a designated endpoint, like an external service. Outbound messages are configured from Setup. You must configure the external endpoint and create a listener for the messages using the SOAP API. You can associate outbound messages with workflow rules, approval processes, or entitlement processes.

For example, automatically initiate the reimbursement process for an approved expense report by triggering an outbound API message to an external HR system.

From Setup, enter *Outbound Messages* in the Quick Find box, and select **Outbound Messages**. Then use these settings to configure your outbound message.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Field	Description
Object	Choose the object that has the information you want included in the outbound message
Name	Enter a name for this outbound message.
Unique Name	Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have outbound messages with the same unique name, provided they are defined for different objects. The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Description	Enter a description that makes it easy for other users to tell what the outbound message does.
Endpoint URL	Enter an endpoint URL for the recipient of the message. Salesforce sends a SOAP message to this endpoint.
User to send as	Select the Salesforce user to use when sending the message. The chosen user controls data visibility for the message that is sent to the endpoint
Send Session ID	Select Send Session ID if you want the Salesforce session ID included in the outbound message. Include the session ID in your message if you intend to make API calls and you don't want to include a username and password. Never send a username and password in an unencrypted message, especially in a production environment. It isn't secure.
Priority	Choose a priority for the outbound message.
Account fields to send	Select the fields to include in the outbound message and click **Add**.

If your endpoint URL uses a client certificate, see Import a Client Certificate for Your Endpoint URL on page 630.

SEE ALSO:

- Track the Delivery Status of an Outbound Message
- Considerations for Outbound Messages
- SOAP API Developer Guide
- Associate Actions with Workflow Rules or Approval Processes
- Considerations for Outbound Messages

Outbound Message Notifications

You can request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request. Failed messages are deleted from the failed outbound messages related list after seven days. Before they are removed, you can delete them yourself or request that they be retried again.

Note: If you don't have this option, your org doesn't have outbound messages enabled. Contact Salesforce to enable outbound messages.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Create an Outbound Message Notification

Request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request.

View an Outbound Message Notification Request

Create an Outbound Message Notification

Request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request.

1. From Setup, enter `Outbound Message Notifications` in the `Quick Find` box, then select **Outbound Message Notifications**.
2. Click **New**.
3. Enter a full username, or click the icon to select it from a list of usernames.
4. Save the request.

Note: If you don't have this option, your org doesn't have outbound messages enabled. Contact Salesforce to enable outbound messages.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create an outbound message notification:

- "Modify All Data"

View an Outbound Message Notification Request

From the detail page of an outbound message notification request:

- Click **Edit** to change the username for a notification request. This is simpler than deleting the request and then creating a new one.
- Click **Delete** to delete the notification request.
- Click **Clone** to create a new notification request with the same username.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view or edit outbound message notification request:

- "Modify All Data"

Track the Delivery Status of an Outbound Message

To track the status of an outbound message, from Setup, enter *Outbound Messages* in the Quick Find box, then select **Outbound Messages**.

- *Next items for delivery* are awaiting delivery.
- *Oldest failures* have not yet been deleted because they have not been delivered and are not 24 hours old.
- *Failed outbound messages* failed to be delivered and are no longer being retried. Messages are listed here only if you configure the message when you create it by selecting Add failures to failed outbound message related list. If you do not see this related list, it has not been enabled for your organization.

You can perform several tasks here.

- Click any workflow or approval process action ID to view the action that triggered it.
- Click **Retry** to change the **Next Attempt** date to now. This causes the message delivery to be immediately retried. If you select **Retry** in the **Failed outbound messages** related list, the outbound message moves to the **Next items for delivery** related list and is retried for another 24 hours.
- Click **Del** to permanently remove the outbound message from the queue.

Note: If you don't have this option, your org doesn't have outbound messages enabled. Contact Salesforce to enable outbound messages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions

Approvals and Milestone Actions are not available in **Database.com**

USER PERMISSIONS

To track outbound messages:

- "Modify All Data"

Import a Client Certificate for Your Endpoint URL

If the endpoint URL of your outbound message uses a client certificate, import it to put your outbound message into action.

1. From Setup, enter *API* in the Quick Find box, then select **API**
2. Click **Generate Client Certificate**.
3. Save the certificate to the appropriate location.

4. Import the downloaded certificate into your application server and configure your application server to request the client certificate.

Define a Flow Trigger for Workflow (Pilot)

Create a flow trigger so that you can launch a flow from workflow rules. With flow triggers, you can automate complex business processes—create flows to perform logic, and have events trigger the flows via workflow rules—without writing code. For example, your flow looks up and assigns the relevant entitlement for a case. Create a flow trigger to launch the flow whenever a case is created, so that all new cases are automatically set with a default entitlement.

Note: The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

To get started using flow triggers, from Setup, enter *Flow Triggers* in the Quick Find box, then select **Flow Triggers**. Before you begin:

- Create and activate the autolaunched flow that you want this workflow action to launch.
- Create the workflow rule that you plan to add this workflow action to.
- Understand the special behavior and limitations of flow triggers. See Flow Trigger Considerations (Pilot) on page 631.

Complete these steps to create a flow trigger.

1. From Setup, enter *Flow Triggers* in the Quick Find box, then select **Flow Triggers**.
2. Click **New Flow Trigger**.
3. Select the same object as the workflow rule, and then click **Next**.
4. Configure the flow trigger.

Field	Description
Name	Name of the flow trigger.
Unique Name	Enter a unique name to refer to this component in the API. The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Protected Component	Reserved for future use.
Flow	Unique name of the autolaunched flow that this workflow action launches.
Set Flow Variables	Whether to pass values into the flow's variables and sObject variables.

5. If you select Set Flow Variables, specify their names and values.

   Click **Set Another Value** to set up to 25 variables.

Field	Description
Name	Select the name of the flow variable or sObject variable. Only variables whose Input/Output Type allow input access can be selected.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view workflow rules and actions:
- "View Setup and Configuration"

To create or change workflow rules and actions:
- "Customize Application"

Field	Description
`Value`	For a flow variable, you can: • Enter a literal value. • Click  , select a field, and click **Insert**. For an sObject variable, you can: • Click  , select an sObject record, and click **Insert**. To help you distinguish between sObject records and fields, all sObject record options are marked with an asterisk (*) and appear at the top of each list. • Enter `{!this}` to use the current values of the record that was created or edited to cause the workflow rule to fire. • Enter `{!old}` to use the most recent previous values of the record that was edited to cause the workflow rule to fire. In other words, `{!old}` identifies the same record as `{!this}` but uses the record's values from immediately before it was edited to cause the workflow rule to fire. **Note:** • If the record was newly created, `{!old}` is `null`. • Unlike `{!this}`, `{!old}` can't be selected by clicking  . You must manually enter `{!old}` in the Value column.

6. To put the flow trigger in test mode, select `Administrators run the latest flow version`.

   When selected and an administrator triggers the workflow rule, the flow trigger launches the latest version of the flow. For all other users, the flow trigger always launches the active version of the flow.

   The same values are passed into the flow variables and sObject variables whether the flow trigger launches the active or latest flow version.

7. Click **Save**.

Don't forget to associate the flow trigger to a workflow rule.

SEE ALSO:

Flow Trigger Considerations (Pilot)

Considerations for Automated Actions

Before you start working with automated actions, familiarize yourself with relevant limits and special behaviors.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Considerations for Field Update Actions

Learn how to use field update actions to their full potential in workflow.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

When creating field updates for workflow rules or approval processes, consider the following:

Field Update Processing

- Field updates occur before email alerts, tasks, and outbound messages.
- Field updates occur after case assignment, lead assignment, and auto-response rules.
- Field updates function independently of field-level security. Therefore, a workflow rule can update fields even though they are hidden on the user's page layout.
- The result of a field update is unpredictable when a single workflow rule includes multiple field updates that apply different values to the same field.
- Field updates may affect the information in a related list. For example, if a field such as the `Amount` or `Close Date` of an opportunity is set to be updated, it will affect the Stage History related list on opportunities.
- If a user gets a field update error when saving a record, you can use the debug log to see which field update failed. The debug log stops when a failure occurs.
- For reminder fields on tasks and events:
  - Field updates can set the reminder for a task or event but they can't use the due date of a task or the scheduled time of an event.
  - Formulas for date/time values are calculated in days. Divide the value by 1440—the number of minutes in a day—to express the value in minutes. For example, the formula `Now()-7` means seven *days* ago, while `Now()-7/1440` means seven *minutes* ago.
- If your organization uses multiple currencies, currency fields are updated using the record's currency. If you choose to update a field based on a formula, any values in your formula are interpreted in the currency of the record.
- Field updates are tracked in the History related list if you have set history tracking on those fields.
- Because updates to records based on workflow rules and also on process scheduled actions don’t trigger validation rules, workflow rules and some processes can invalidate previously valid fields.
- If you have person accounts enabled, you can use the `Is Person Account` field as part of the evaluation criteria for workflow rules. However, because the `Is Person Account` field is read-only, any field updates set up to modify it will fail.

Tip: Salesforce processes rules in the following order:

1. Validation rules
2. Assignment rules
3. Auto-response rules
4. Workflow rules (with immediate actions)
5. Escalation rules

Notes on Cross-Object Field Updates

- For all custom objects and some standard objects, you can create workflow and approval actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships, custom-to-standard master-detail relationships, and a few standard-to-standard master-detail relationships. For more information, see Cross-Object Field Updates on page 11.
- Approval processes can't use cross-object field update actions.
- An approval process can specify a field update action that reevaluates workflow rules for the updated object. If, however, the re-evaluated workflow rules include a cross-object field update, those cross-object field updates are ignored.
- To create workflow rules so that case comments or emails automatically update fields on associated cases, select **Case Comment** or **Email Message** in the Object drop-down list when creating a new workflow rule and select **Case** in the Field to Update list. Email-to-Case or On-Demand Email-to-Case must be enabled for your organization to use the Email Message in a workflow rule.

  When cases are updated by an email-triggered workflow rule, the updated case can trigger:

  - Workflow rules
  - Validation rules
  - Updates to roll-up summary fields
  - Escalation rules
  - Apex triggers
  - Entitlement processes

  The updated case can't trigger:

  - Assignment rules
  - Auto-response rules

Field Update Actions and Custom Fields

- Before changing a custom field's type, make sure that it isn't the target of a workflow field update or referenced in a field update formula that would be invalidated by the new type.
- You can't delete a custom field that is referenced by a field update.
- You can use field updates on encrypted custom fields, but if you try to use a formula to set the new value, the encrypted field isn't available in the formula editor.

Field Update Actions on Opportunities and Contracts

- You can define field updates for the `Stage` field on opportunities, but be aware of how this field affects the `Type` and `Forecast Category` fields.

- You can define field updates using the `Amount` field on opportunities but it will only apply to those opportunities that don't have products. Adding products to an opportunity changes the `Amount` field to a read-only field that is automatically calculated and not affected by that field update.
- You can define field updates for the `Status` field on contracts. However, the value of this field may affect the value of the `Status Category` field as well.
- Avoid creating a field update for contracts or orders that changes the `Status` field to any value other than Approved.

Field Update Action Limitations

- The results of a field update can't trigger additional rules such as validation, assignment, auto-response, or escalation rules.
- The results of a field update can trigger additional workflow rules if you have flagged the field update to do so. For more information, see Field Updates That Re-evaluate Workflow Rules on page 630.
- Field updates that are executed as approval actions don't trigger workflow rules or entitlement processes.
- These fields aren't available for field update actions:
  - Read-only fields like formula or auto-number fields
  - The `Language` picklist field on multilingual solutions
  - Some activity fields, such as `Related To` and `Private`
- Email message workflow rules can only be associated with field updates.
- If a field update references a specific user, you can't deactivate that user. For example, if your field update is designed to change the owner of a record to Bob Smith, change the field update before deactivating Bob Smith.
- You can update long text area fields, but the option to insert `A specific value` restricts you to entering up to the maximum amount of characters allowed in the destination field.
- You can't make a field universally required if it's used by a field update that sets the field to a blank value.
- Workflow rules that update owners *do not* also transfer associated items. To ensure transfer, click **Change** next to the owner's name in a record and make your transfer selections.

IN THIS SECTION:

Cross-Object Field Updates

For all custom objects and some standard objects, you can create actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships, custom-to-standard master-detail relationships, and a few standard-to-standard master-detail relationships.

Field Updates That Re-evaluate Workflow Rules

If `Re-evaluate Workflow Rules After Field Change` is enabled for a field update action, Salesforce re-evaluates all workflow rules on the object if the field update results in a change to the value of the field.

SEE ALSO:

Cross-Object Field Updates

Cross-Object Field Updates

For all custom objects and some standard objects, you can create actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships, custom-to-standard master-detail relationships, and a few standard-to-standard master-detail relationships.

For example, in a custom recruiting application, create a workflow rule that sets the status of an application (the master object) to "Closed" when a candidate (the detail object) accepts the job. Or, for standard objects, create a rule to change the status of a case from "Awaiting Customer Response" to "In Progress" when a customer adds a case comment.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Custom Object to Custom Object

Cross-object field updates are supported for all custom objects that are children of custom objects in a master-detail relationship.

Custom Object to Standard Object

Cross-object field updates are supported for custom objects that are children of certain standard objects in a master-detail relationship. The standard objects which support cross-object field updates from custom objects are:

- Account
- Asset
- Campaign
- Case
- Contact
- Contract
- Contract Line Item
- Entitlement
- Opportunity
- Order
- Question
- Quote
- Service Contract
- Solution

Standard Object to Standard Object

Cross-object field updates are supported for standard objects that are children of standard objects in a master-detail relationship. However, only these standard-to-standard relationships are supported.

- Case Comments updating Case
- Email updating Case

  Tip: To create workflow rules so that case comments or emails automatically update fields on associated cases, select **Case Comment** or **Email Message** in the Object drop-down list when creating a new workflow rule and select **Case**in the Field to Update list. Email-to-Case or On-Demand Email-to-Case must be enabled for your organization to use the Email Message in a workflow rule.

- Opportunity Product updating Opportunity

Note: Cross-object field updates to a parent opportunity's `Amount` and `Quantity` fields only work if the opportunity has no opportunity products associated with it.

- Opportunity updating Account—Supported for both business accounts and person accounts.

Note: If you have workflow rules on converted leads and want to use cross-object field updates on the resulting accounts and opportunities, you must enable the lead setting `Require Validation for Converted Leads`.

Standard-to-standard cross-object field update actions:

- Can't be used in, or assigned to, approval processes.
- Update a parent record even if the user doesn't have edit access to it.

Note: If you have Apex code that updates parent fields in the same relationships as a cross-object field update action, consider replacing your code with cross-object field updates. Otherwise, both will fire, and since workflow rules run after Apex triggers, the workflow field update will override any change made by your Apex code.

SEE ALSO:

Considerations for Field Update Actions

Object Relationships Overview

Field Updates That Re-evaluate Workflow Rules

If `Re-evaluate Workflow Rules After Field Change` is enabled for a field update action, Salesforce re-evaluates all workflow rules on the object if the field update results in a change to the value of the field.

- If the field update changes the field's value, all workflow rules on the associated object are re-evaluated. Any workflow rules whose criteria are met as a result of the field update will be triggered.
- If any of the triggered workflow rules result in another field update that's also enabled for workflow rule re-evaluation, a domino effect occurs, and more workflow rules can be re-evaluated as a result of the newly-triggered field update. This cascade of workflow rule re-evaluation and triggering can happen up to five times after the initial field update that started it.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

- Make sure that your workflow rules aren't set up to create recursive loops. For example, if a field update for Rule1 triggers Rule2, and a field update for Rule2 triggers Rule1, the recursive triggers may cause your organization to exceed its limit for workflow time triggers per hour.
- In a batch update, workflow is only retriggered on the entities where there is a change.
- Only workflow rules on the same object as the initial field update will be re-evaluated and triggered.
- Only workflow rules that didn't fire before will be retriggered.
- Cross-object workflow rules aren't candidates for re-evaluation.
- Cross-object field updates that cause a field value to change don't trigger workflow rule re-evaluation on the associated object.
- An approval process can specify a field update action that reevaluates workflow rules for the updated object. If, however, the re-evaluated workflow rules include a cross-object field update, those cross-object field updates are ignored.
- Time-dependent actions aren't executed for a reevaluated workflow rule in the following situations:
  - The reevaluated workflow rule's immediate actions cause the record to no longer meet the workflow rule criteria.

- An Apex `after` trigger that is executed as a result of a workflow or approvals action causes the record to no longer meet the workflow rule criteria.

SEE ALSO:

Considerations for Field Update Actions

Considerations for Outbound Messages

Review the considerations for using outbound message actions before implementing them in your workflows.

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

When creating outbound messages for workflow rules or approval processes, consider the following:

- A single SOAP message can include up to 100 notifications. Each notification contains an ID that uniquely identifies a record, and a reference to the data in the record. Therefore, if the information in the record changes after the notification is sent, but before the notification is delivered, only the updated information is delivered.
- Messages are queued until they are sent, to preserve message reliability.
- If the endpoint is unavailable, messages will stay in the queue until sent successfully or until they are 24 hours old. After 24 hours, messages are dropped from the queue.
- If a message can't be delivered, the interval between retries increases exponentially, up to a maximum of two hours between retries.
- Messages are retried independent of their order in the queue. This may result in messages being delivered out of order.
- You can't build an audit trail using outbound messages. While each message should be delivered at least once, it may be delivered more than once. Also, it may not be delivered at all if delivery cannot be done within 24 hours. Finally, as noted above, the source object may change after a notification is sent but before it is delivered, so the endpoint will only receive the latest data, not any intermediate changes.

Flow Trigger Considerations (Pilot)

Flow trigger workflow actions have special behaviors and limitations.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: The pilot program for flow trigger workflow actions is closed. If you've already enabled the pilot in your org, you can continue to create and edit flow trigger workflow actions. If you didn't enable the pilot in your org, use the Flows action in Process Builder instead.

Understand these considerations before you create flow triggers or add them to workflow rules.

- Flow triggers are available only for workflow rules. You can't use them as actions elsewhere, for example, in approval processes.
- Flow triggers are available on most—but not all—objects that are supported by workflow rules. You can see the list of supported objects when you create a new flow trigger. From Setup, enter *`Flow Triggers`* in the `Quick Find` box, then click **Flow Triggers**.
- Only active, autolaunched flows can be launched by flow triggers. However, if a flow trigger is in test mode, administrators run the latest flow version while other users run the active flow version.
- Flows that are launched from workflow rules are run in system context, which means that user permissions, field-level security, and sharing rules aren't taken into account during flow execution.
- If a flow trigger fails at run time, the user who created or edited the record to meet the workflow rule criteria won't be able to save the record. To troubleshoot run time issues, see the flow action events in the `Workflow` category of debug logs, which show the flow version and the values passed into flow variables.

- A flow trigger can set the values of up to 25 variables and sObject variables in the flow, with the following limitations.
  - Flow triggers can't use multi-select picklist fields to set flow variables or sObject variables.
  - When a flow trigger uses a currency field to set a flow variable, only the amount is passed into the flow. Any currency ISO code or locale information is ignored. If your organization uses multiple currencies, the flow trigger uses the amount in the currency of the record that contains the specified currency field.
  - Flow triggers can't pass values into sObject collection variables in flows.
- Always keep one version of the flow active if it's referenced by an active workflow rule's flow trigger.
- Once you activate a workflow rule using the flow trigger, don't modify or add a version of the flow to include screens or other elements that would violate the run restrictions for an autolaunched flow. If you modify a flow to no longer be autolaunched, it can't be launched by flow triggers. To work around this situation, you can save the non-autolaunched flow as a new flow and change the new flow to become autolaunched. Then update the flow triggers to launch the new flow.
- Flow triggers aren't available as time-dependent workflow actions. You can add flow triggers to workflow rules only as immediate workflow actions.
- When the system executes a workflow rule with multiple flow triggers, those flows aren't run in any particular order.
- In a transaction, flow triggers are executed after all workflow field updates, including any Apex triggers and standard validations that are executed as a result of those workflow field updates. After executing flow triggers, the system executes escalation rules.
- Flows that are launched from workflow rules are governed by the per-transaction limits already enforced by Apex.
- When flows are launched from workflow rules that are triggered by bulk loads or imports, the flows' data manipulation language (DML) operations are executed in bulk to reduce the number of calls required and to optimize system performance. The execution of any of the following flow elements qualifies as a DML operation: Record Create, Record Update, Record Delete, Fast Create, Fast Update, or Fast Delete.

  For example, suppose that you use Data Loader or the Bulk API to update 50 records, and those updates meet the criteria of a workflow rule with a flow trigger action. In response, the system executes 50 instances of the flow within the same transaction. Each instance of a running flow is called an interview. The system attempts to execute each DML operation across all the interviews in the transaction at the same time. Suppose that five of those interviews are executing the same branch of the flow, which has a Record Update element called "SetEntitlement." The system waits for all five interviews to reach that element, and then executes all five record updates in bulk.
- Flow triggers aren't available in change sets.
- Flow triggers aren't packageable.

Update Process Automation Settings

To use time-based workflow, let users respond to approval requests via email, let users to pause flow interviews, or enable Lightning runtime for flows, update your org's process automation settings.

From the Process Automation Settings page:

- Set the `Default Workflow User` for your Salesforce org
- Enable your users to respond to approval requests via email
- Display the **Pause** button on flow screens
- Enable Lightning runtime for all URL-based flows

EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit process automation settings:

- "Customize Application"

Creating Websites

Domain Management

Domain Management Overview

The Domain Management page allows you to configure domains and their site associations with a high degree of flexibility.

Sites and domains can have a many-to-many relationship. Each domain can have up to 50 sites, and each site can be associated with up to 100 domains. For example, you might want to set up a single domain to host your Communities, Force.com, and Site.com sites. This simplifies your domain requirements. For Site.com sites, you might want to host them on the same domain as your Force.com sites because you can access Visualforce pages and have easier access to Apex code.

There are also reasons why you would have a site on more than one domain. For example, let's say you have a parent company with two distinct brands. Each brand has its own registered domain, but you want them both to point to the parent website. Because you can have a site exist on more than one domain, you can point both brand domains to a single parent website.

If you plan to host more than one site on a domain, you'll need to set up custom URLs for each site. Custom URLs are the way to uniquely distinguish the sites within that domain. Let's say you have a domain called `www.ourdomain.com` and you want to host two sites called `siteone` and `sitetwo`. You'll need to create custom URLs by associating ourdomain.com to each site using a custom path. This will result in two custom URLs: `http://www.ourdomain.com/siteone`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To manage domains:

- You must have Site.com, Force.com Sites, or Communities enabled

and `http://www.ourdomain.com/sitetwo`. When a web user accesses the domain using one of the URLs, the custom path determines which site within the domain they see.

SEE ALSO:

- Adding a Domain
- Adding a Custom URL
- Managing Domains and Custom URLs
- Deleting a Domain
- Deleting Custom URLs

Adding a Domain

You can add domains and attach certificates using the Domain Management page in Setup.

When you add a domain, you also have the option of attaching a certificate and key for connection security. If you're going to use a certificate, make sure you've already added it using Certificate and Key Management before you try to attach it to your domain. Only CA-signed certificates are supported, and they must be 2048 bits in length. To support all domains that are hosted by sites in your organization, use a wildcard or Subject Alternative Name certificate.

Custom domains are supported only in non-sandbox instances. You can configure a custom domain in a sandbox instance and then migrate it to a production instance, but the custom domain is only active in production.

To add domains in your organization:

1. From Setup, enter *Domains* in the `Quick Find` box, then select **Domains**.
2. Click **Add a Domain**.
3. Enter the Domain Name.
4. Add a certificate if you have already set up a CA-signed certificate that supports this domain.
5. Click **Save**. Alternatively, click **Save & New** to add multiple domains.

Before you switch the CNAME of your domain name to point to a new target name, ensure that the new target name exists in the DNS by using `dig` or `nslookup`. When you created your domain names affects the target of your CNAME:

- Domain names that were added before Summer '13, typically need to have their CNAME adjusted to point to the fully qualified domain followed by `.live.siteforce.com` instead of to the organization's `force.com` sub-domain. For example, if your pre-Summer '13 domain is `www.example.com`, then the target of its CNAME will need to be `www.example.com.live.siteforce.com` instead of `example.force.com` before HTTPS will work.
- Domain names that were added in or before Summer '13, don't have the 18-character organization ID in the CNAME target.
- Domain names that were added in or after Summer '13, already point to the proper place for setting up HTTPS in a custom domain.
- Domain names that were added in or after Winter '14, use a CNAME that points to the fully qualified domain followed by your organization's 18-character ID and `.live.siteforce.com`. For example, if your domain name is `www.example.com`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To view domains:
- "View Setup and Configuration"

To add domains:
- "Customize Application" or "View Setup and Configuration" plus either a Site.com Publisher license or "Create and Set Up Communities"

To edit or delete domains:
- "Customize Application"

To associate certificates with a domain:
- Contact Salesforce if you aren't using Communities. Organization with Communities can associate certificates with a domain.

and your 18-character organization ID is `00dxx0000001ggxeay`, then the target of its CNAME will need to be `www.example.com.00dxx0000001ggxeay.live.siteforce.com`.

SEE ALSO:

- Domain Management Overview
- Adding a Custom URL
- Managing Domains and Custom URLs
- Deleting a Domain
- Deleting Custom URLs

Adding a Custom URL

Once you've added domains to the Domain Management page, you can select your domain and site relationships by creating a custom URL. A custom URL consists of the domain and a custom path. The same path name can be used on more than one domain, but it can't be used more than once within the same domain. When adding a custom URL, the / is required and indicates the root. You can add an additional path after the /, but you must at least use the / to indicate the root. For example, if the domain name is `https://oursite.com` and the path is `/products`, the site URL is `https://oursite.com/products`. If you added the custom URL to the root, the URL would be `https://oursite.com`.

If you want to set a preferred custom URL for authenticated pages and emails that links back to the site or community, then select **Site Primary Custom URL**. This is only available on the root path for Force.com and Communities sites and not available at all on Site.com sites. For Chatter Communities, if you don't select a primary URL, then the first `https` custom domain in the site is used for authenticated pages and emails. If there is no custom `https` domain, then your Force.com domain is used.

To add a custom URL:

1. From Setup, enter *`Custom URLs`* in the `Quick Find` box, then select **Custom URLs**. Alternatively, enter *`Domains`* in the `Quick Find` box, select **Domains**, and click the domain name.
2. Click **New Custom URL**.
3. Enter a Domain name.
4. Enter a Site name.
5. Enter a unique path.
6. Click **Save**. Alternatively, click **Save & New** to add multiple Custom URLs.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To view custom URLs:

- "View Setup and Configuration"

To add, edit, and delete custom URLs:

- "Customize Application" or "View Setup and Configuration" plus either a Site.com Publisher license or "Create and Set Up Communities"

Example: In the Custom URLs table, even though the domain name and path show in separate columns in the table, it is the combination that make up the actual URL.

SEE ALSO:

- Domain Management Overview
- Adding a Domain
- Managing Domains and Custom URLs
- Deleting a Domain
- Deleting Custom URLs

Managing Domains and Custom URLs

You can use Domain Management in Setup to manage all the certificates, custom URLs, and sites attached to a domain.

You might want to add a new custom URL to an existing domain, rename a domain, or delete a custom URL from a particular domain. The Domain Detail page allows you to manage the domains and URLs from one page. The top section contains information about the domain and certificates. The lower section contains the a Custom URL related list for the domain. Using this page you can:

- Edit the domain name and certificate — click **Edit** in the Domain Detail section.
- Delete the domain — click **Delete** in the Domain Detail section.
- Add new custom URLs — Click **New Custom URL**
- Edit existing custom URLs — Click **Edit** on a custom URL row.
- Delete custom URLs — Click **Del** on a custom URL row.
- View the site if it is published — Click **View** on a custom URL row.
- Jump to a site — Click the site name below the Site Label.
- Jump to an attached certificate — click the certificate name next to Certificate and Key.
- See whether the domain has External HTTPS enabled. Only domains that don't point to the *yourdomain.your18characterOrgId*`.live.siteforce.com` CNAME target have this option available.

Note: You can't edit a domain name that is used by a community.

To access the Domain Detail page:

1. From Setup, enter *Domains* in the `Quick Find` box, then select **Domains**.
2. From the Domain Name column, click the domain name.

Note: If you have your domain set to use `http`, and you edit the domain to use `https` instead, you must republish your Site.com sites.

SEE ALSO:

Domain Management Overview

Adding a Custom URL

Deleting Custom URLs

Enable External HTTPS on a Domain

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To view domains:
- "View Setup and Configuration"

To add domains:
- "Customize Application" or "View Setup and Configuration" plus either a Site.com Publisher license or "Create and Set Up Communities"

To edit or delete domains:
- "Customize Application"

To associate certificates with a domain:
- Contact Salesforce if you aren't using Communities. Organization with Communities can associate certificates with a domain.

USER PERMISSIONS

To view custom URLs:
- "View Setup and Configuration"

To add, edit, and delete custom URLs:
- "Customize Application" or "View Setup and Configuration" plus either a Site.com Publisher license or "Create and Set Up Communities"

Deleting a Domain

You can delete domains from your organization using the Domain Management page.

You can't delete a domain that is attached to a published site. You'll need to unpublish all sites attached to the domain before you can delete it. For Force.com sites, the domain is deleted immediately, and users no longer have access to any site connected to that domain. But with Site.com sites, the domain remains active until you republish the site.

To delete a domain from your organization:

1. From Setup, enter *Domains* in the Quick Find box, then select **Domains**.
2. Click **Del** next to the domain name.
3. Click **OK**.

When you delete a domain with an unpublished site attached, it also deletes the custom URL associated with that site.

SEE ALSO:

Domain Management Overview

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To delete domains:

- "Customize Application"

Deleting Custom URLs

You can delete custom URLs from your organization using the Domain Management page.

You can't delete a URL that is attached to a published site. You'll need to unpublish the site attached to the custom URL before you can delete it. For Force.com sites, the URL is deleted immediately, and users no longer have access to any site connected to that URL. But with Site.com sites, the URL remains active until you republish the site.

To delete a URL from a domain:

1. From Setup, enter *Custom URLs* in the Quick Find box, then select **Custom URLs**.
2. Click **Del** next to the URL name.
3. Click **OK**.

SEE ALSO:

Domain Management Overview

Managing Domains and Custom URLs

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To view custom URLs:

- "View Setup and Configuration"

To add, edit, and delete custom URLs:

- "Customize Application" or "View Setup and Configuration" plus either a Site.com Publisher license or "Create and Set Up Communities"

Enable External HTTPS on a Domain

Enabling External HTTPS on a domain allows Salesforce to use the domain in secure references to the sites that are associated with that domain, such as in URLs that appear within an email message from a Salesforce Community. This feature is useful for sites that require authentication, such as community sites, portal sites, or any type of site where confidentiality, protection from eavesdropping, or protection against spoofing are important.

This option is intended for domains that use a non-Salesforce content delivery network (CDN) or endpoint to serve HTTPS.

You can enable External HTTPS on any domain that doesn't point to the *yourdomain.your18characterOrgId*`.live.siteforce.com` CNAME target.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions

When enabled, and when the domain's DNS record is not a CNAME that points to the `live.siteforce.com` subdomain for that domain, Salesforce assumes that the domain supports secure connections.

When disabled or when the domain's DNS record is a CNAME that points to the expected `live.siteforce.com` subdomain for that domain, Salesforce assumes that the domain doesn't support secure connections.

You can't enable External HTTPS when you add a new domain, only when editing an existing one.

To enable External HTTPS on a domain, click **Edit** from the domain list page or from a domain's detail page, and then select `Enable External HTTPS`.

If the domain that you're enabling External HTTPS on comprises only noncommunity Site.com custom URLs, you must manually publish any affected Site.com sites via the Site.com Studio to have this change take effect for those sites. If at least one community site or at least one Force.com site exists on the same domain as a non-Community Site.com site, the publishing of changes to this Enable External HTTPS bit is automatic on that domain.

SEE ALSO:

- Managing Domains and Custom URLs
- Domain Management Overview

Community Builder (Site.com) and Force.com Sites Overview

Community Builder (Site.com)

Community Builder is an intuitive, convenient tool for customizing your community. Community Builder lets you create a community based on a preconfigured template, and then apply branding, edit pages, update your template, and publish changes all from one user-friendly interface.

Here's what you can achieve with Community Builder:

- Use one of the self-service templates—Koa, Kokua, or Customer Service (Napili)—to easily create a responsive self-service community.
- Use the Aloha template to create a configurable App Launcher.
- Design pixel-perfect, branded pages.
- Create public pages that anyone can access, or add private pages that you can add as a tab within your community.
- Build and iterate quickly using drag-and-drop reusable page elements.
- Use ready-made forms to create web-to-lead forms or gather customer feedback.
- Create data-driven pages, such as product catalogs or other listings, using your organization's data.

- Set a home page and set up multilingual support for your community in Site.com Studio, a Web content management system that provides extra configuration options. Site.com Studio is easily accessible from Community Management.

Force.com Sites

Suitable for developers with experience using Visualforce, Force.com Sites lets you build custom pages and Web applications by inheriting Force.com capabilities including analytics, workflow and approvals, and programmable logic. So if you are looking to create sites programmatically using Apex and APIs, Force.com Sites is the product for you. Here's what you can achieve with Force.com:

- Create public, branded pages that anyone can access.
- Leverage Visualforce to create private pages that you can add as a tab within your community.
- Write your own controllers, or extensions to controllers, using Apex code.
- Create custom login or self-registration pages.
- Build dynamic web applications, such as an event management application.

Features At a Glance

Still unsure which product to choose? Take a look at this table to learn more about each product's features.

Feature	Community Builder	Force.com Sites
Public pages	✓	✓
Community templates (Koa, Kokua, Customer Service (Napili), Aloha)*	✓	
Authenticated pages*	✓	✓
Visualforce pages		✓
Out-of-the-box login, logout, self-registration, and error pages	✓	✓
Drag-and-drop environment	✓	
Reusable components	✓	
Pixel-perfect designs	✓	✓
IP restrictions	✓	✓
Access to data, such as cases, leads, and opportunities	✓	✓
Ready-made forms	✓	
CMS	✓	
Programmatic page creation (using Apex, APIs, and controllers)		✓
Web applications		✓
Analytics, reports, and workflows		✓

Feature	Community Builder	Force.com Sites
Full Force.com platform capability		✓

*Available in Community Builder for Communities users only.

Site.com

Site.com Overview

Site.com is a Web content management system (CMS) that makes it easy to build dynamic, data-driven Web pages quickly, edit content in real time, and manage your websites. From the Site.com tab in the Site.com app, you can launch Site.com Studio, which provides a separate, dedicated environment for creating and editing pixel-perfect, custom websites. Site administrators and designers can create and style Web pages, and add features such as navigation menus, images, and text areas using drag-and-drop page elements, while ensuring the site's pages match the look and feel of the company's brand. And content contributors, such as marketing users, can browse and update website content directly in a simplified Site.com Studio environment. Additionally, websites built with Site.com benefit from running on Salesforce's trusted global infrastructure.

Note: The features available in Site.com Studio vary depending on whether you're a site administrator, designer, or contributor.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

The following examples illustrate a few ways to use Site.com:

- Create an event site—Advertise upcoming events, such as grand openings, launches, or sales kick-offs on a public event site.
- Promote new products—Launch new products into the market with a promotional website that helps drive sales.
- Publish a support FAQ—Provide helpful information on a public website where customers can view solutions to their issues.
- Create microsites and landing pages—Create temporary landing pages or targeted microsites for marketing campaigns.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes.
- Publish a catalog of products—List all of your company's products on a public website, with model numbers and current prices pulled dynamically from your organization.
- Post company press releases—Publish your company's press releases and sort by publication date.

System Requirements

To use Site.com Studio, we recommend:

- Mozilla® Firefox® or Google® Chrome for best performance. Windows® Internet Explorer® versions 6.0 and 7.0 are not supported.
- Disabling the Firebug extension for Firefox, if installed, as it can impact performance.

- A minimum browser resolution of 1024x768.

SEE ALSO:

Setting Up Site.com Users

Planning and Implementing a Site.com Website

Using Site.com Studio as a Site Administrator or Designer

Using Site.com Studio as a Contributor

Setting Up Site.com Users

About Site.com Feature Licenses

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

To access Site.com, each user in your organization must have a Site.com feature license.

- The Site.com Publisher feature license allows the user to access Site.com Studio to create and style websites, control the layout and functionality of pages and page elements, and add and edit content.
- The Site.com Contributor feature license allows the user to access Site.com Studio to edit site content only.

Consider what your users need to do in a site and purchase feature licenses accordingly. See the Site.com feature table for a complete list of the capabilities that come with each feature license. After you purchase feature licenses, you can set up Site.com users.

You can view the number of assigned feature licenses on your organization's profile.

Note: Organizations using Performance, Unlimited, or Enterprise Editions must purchase Site.com Publisher and Site.com Contributor feature licenses separately. Additionally, a Site.com Published Site license is required for each site that's published to the Internet. For information on purchasing Site.com licenses, contact Salesforce.

Developer Edition organizations contain two Site.com Publisher feature licenses and one Site.com Contributor feature license. Developer Edition organizations can't publish sites.

Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. To let Communities users without the permission edit the site, you must purchase and assign either a Site.com Publisher or a Site.com Contributor feature license, and assign a user role at the site level.

SEE ALSO:

About Site.com User Roles

Managing Site.com Users and Roles

Setting Up Site.com Users

Before users can access Site.com, you must allocate a Site.com feature license to each user and assign a user role at the site level. The features available when editing a site in Site.com Studio vary depending on these settings.

First, review the Site.com feature table for detailed information on the feature license, permissions, and role required for each user.

After you've determined the appropriate access level required:

1. Allocate a feature license to the user by editing the user's profile. To allocate:
   - A Site.com Publisher feature license, select the `Site.com Publisher User` checkbox.
   - A Site.com Contributor feature license, select the `Site.com Contributor User` checkbox.

   After you allocate a feature license, users can access Site.com in the Force.com app menu in the Salesforce header.

   Note: If the checkboxes don't appear, verify that Site.com is enabled for your organization. See About Site.com Feature Licenses on page 642.

2. Ensure the "View Setup and Configuration" permission is enabled. All users who create or edit websites in Site.com Studio need this permission.
3. Additionally, ensure that at least one user in your organization has both a Site.com feature license and the "Manage Users" permission. This way, someone can always reallocate user roles if a site's users are accidentally deleted.

   Warning: The "Manage Users" permission is powerful. It allows a user to manage all other users in your organization and not just Site.com.

4. Add users and assign user roles within a site. (When a user with the Site.com Publisher feature license creates a site, the user is automatically allocated the role of site administrator at the site level.)

The feature license, permissions, and user role all determine what a user can do in each site. For example, to create an administrative user who can manage all sites in your organization, assign a Publisher feature license and assign the role of site administrator at the site level.

For users who need only limited access to edit site content, but no administrative access, assign a Contributor feature license and a contributor role at the site level.

Alternatively, to create a user who can manage roles in a site, but without the ability to publish, assign a Publisher feature license, the "Manage Users" permission, and the designer role at the site level.

SEE ALSO:

About Site.com User Roles

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To create or edit users:
- "Manage Internal Users"

About Site.com User Roles

Each Site.com user must have a user role assigned at the site level, which controls what each user can do in a site. Users can have only one role per site, but their roles can vary between sites. For example, a person can be a site administrator on one site and a contributor on another.

To manage user roles in a site, you must either be the site administrator for that site, or have a Site.com feature license and the "Manage Users" permission.

Note: Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. However, they don't appear in the User Roles section on the Overview tab of Site.com Studio.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Users can have one of three roles at the site level:

- Site administrator—Site administrators are users who can create and manage all site content. They can create sites, templates, style sheets, and pages, and also set up domains, publish sites, and assign user roles. This role requires the Site.com Publisher feature license.
- Designer—Designers have the same control over site content as site administrators, but they can't manage domains or publish sites. By default, they can't assign roles unless they have the "Manage Users" permission. This role requires the Site.com Publisher feature license.
- Contributor—Contributors have the most restricted access to content and can typically just edit page text and images. By default, they can't assign roles unless they have the "Manage Users" permission. This role requires the Site.com Contributor feature license.

See the Site.com feature table on page 646 for a detailed list each user role's capabilities.

SEE ALSO:

Managing Site.com Users and Roles
Setting Up Site.com Users
Setting Up the Contributor's Studio View
About Site.com Feature Licenses

Managing Site.com Users and Roles

After you create a site, you can add other users and assign roles to them. If you haven't assigned Site.com feature licenses to your users, you won't be able to add them to a site.

Note: Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. However, they don't appear in the User Roles section on the Overview tab of Site.com Studio.

When assigning a user role, be sure to add one that's compatible with the user's Site.com license. When users log into Site.com, their licenses are checked against the role assigned to them at the site level. If the license doesn't allow the permissions associated with the role, then the user is given the permissions associated with the license. For example, if a user has a Site.com Contributor feature license, but is assigned a role of site administrator, they will only have Contributor permissions regardless of the assigned role.

To add users and assign roles:

1. On the Overview tab in Site.com Studio, click **Site Configuration** > **User Roles**.
2. Click **Add Users**.
3. In the Available Users section, highlight the user you want to add.
4. Select the role from the Add as drop-down list.
5. Click the arrow to move the user to the Selected Users section.
6. Click **Save**.

To delete users:

1. In the User Roles view, select the user.
2. Click > **Remove**.
3. Click **OK**.

To change a user's role:

1. In the User Roles view, hover over the user's role.
2. Click the arrow to display all the roles.
3. Select the new role.

To delete or change the role of a group of users at the same time, use Bulk Actions.

1. In the User Roles view, select the check box beside each user's name.
2. Click **Bulk Actions**.
3. Select the action.
4. Click **Apply**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage Site.com users:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Note: When updating the roles of several users at once, you can only assign the same role to all selected users.

SEE ALSO:

- Setting Up Site.com Users
- Setting Up the Contributor's Studio View
- Planning and Implementing a Site.com Website
- About Site.com Features

About Site.com Features

The features available when editing a site in Site.com Studio vary depending on your Site.com feature license and also your user role for that particular site.

This table lists the required feature licenses, permissions, and roles for many of the Site.com Studio features.

Note:

- Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site.
- You can't create, delete, or duplicate community sites in Site.com.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Site.com Studio Feature Requirements

Feature	Feature License	Site.com Studio User Role	Permissions
Assign feature license to user profile			"Manage Internal Users"
Add users and roles at the site level	Publisher	Site administrator	Additionally, any user with a Site.com feature license and the "Manage Users" permission.
Enable contributors to create pages, add content blocks and widgets, and edit content blocks and graphics	Publisher	Site administrator or designer	
Create websites	Publisher	Users who create a site are automatically added to that site as a site administrator.	
Delete websites	Publisher	Site administrator or designer	
Import websites	Publisher	Users who import a site are automatically added to the	

Site.com Studio Feature Requirements

Feature	Feature License	Site.com Studio User Role	Permissions
		new site as a site administrator.	
Export websites	Publisher	Site administrator or designer	
Duplicate websites	Publisher	Site administrator or designer	
Manage domains (Unavailable for Developer Edition)	Publisher	Site administrator	
Add and edit IP restrictions	Publisher	Site administrator	
Publish changes to the live website (Unavailable for Developer Edition)	Publisher	Site administrator	
Create page templates	Publisher	Site administrator or designer	
Create website pages	Publisher or Contributor	Site administrator or designer Contributor only if enabled by the site administrator or designer in the page template's Properties pane.	
Create and modify style sheets	Publisher	Site administrator or designer	
Modify layout and design	Publisher	Site administrator or designer	
Add page elements	Publisher or Contributor	Site administrator or designer Contributor can add content blocks and widgets only if enabled by the site administrator or designer.	
Add data repeaters and other data-bound page elements	Publisher	Site administrator or designer	
Modify the Guest User profile to set public access permissions to Salesforce objects	Publisher	Site administrator or designer	"Manage Profiles and Permission Sets" and "Customize Application"
Import assets, such as images and files	Publisher or Contributor	Any assigned role	
Edit content and images	Publisher or Contributor	Site administrator or designer Contributor only if enabled by the site administrator or designer in the page template's Properties pane.	
Preview website pages	Publisher or Contributor	Any assigned role	

Getting Started with Site.com

Planning and Implementing a Site.com Website

There are many approaches to building a website. The process that best suits you depends on many factors, such as the size of your team and the tasks you're responsible for.

If you're a site administrator or designer, you may be involved in every stage, including adding and maintaining the site's content. Alternatively, you may have contributors who add, edit, and maintain this content. And if you're a contributor, you may be responsible for editing and updating all of the site's content, or you may work with other contributors, designers, and site administrators to bring the site to completion.

This topic describes the various stages involved in creating a site with Site.com.

- **Plan the Site Design and Page Layout** (Site administrator or designer)—Before building the pages of the site, spend time planning the site design and basic layout. This stage is key to ensuring a consistent look and feel with the minimum amount of effort. From a hierarchical point of view, think about how many pages you need and whether they'll have subpages. Also consider how you want site visitors to navigate around your site.

  Next, plan the layout of the pages and identify the common elements that every page will have. In this example, the site has a header section that includes the company's logo and menu (1), and a footer section (2). However, the main section of the home page (3) differs from the rest of the site pages (4). Take note of these similarities and differences, because they will affect how you create your site pages.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To create or import Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

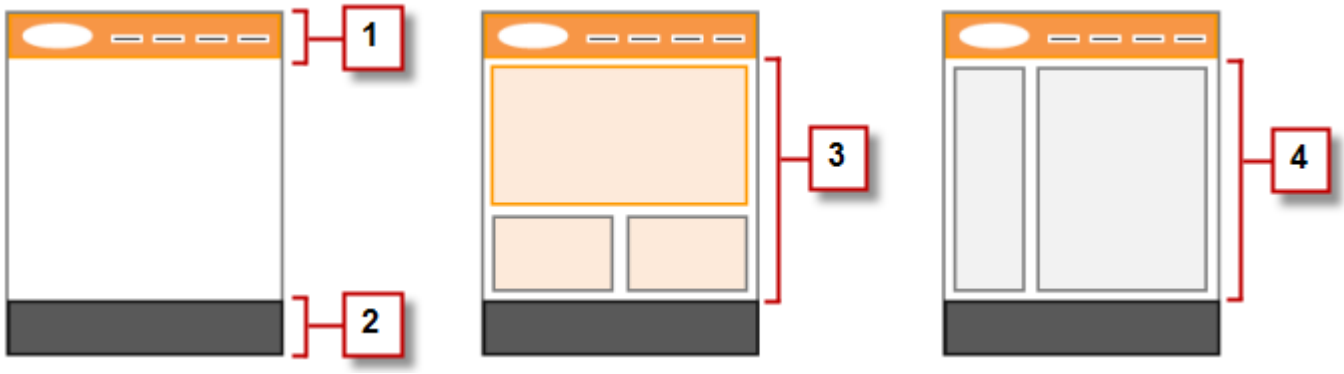
  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level



- **Create the Site** (Site administrator or designer)—Once you've completed the planning stage, you're ready to get started! Log into the Site.com app and go to the Site.com tab, where you can create your first site. Your new blank site opens in Site.com Studio, a powerful environment for building the pages of your site.

  Note: Only users with the `Site.com Publisher User` field enabled on their user detail page can create and import sites.

- **Import Assets** (Site administrator or designer)—If you're working with a design agency, they may provide all of the files and assets you need, including a CSS style sheet. If you've created your own design, cut up the design and collect the assets, images, and files you plan to use in the site. Import the assets into Site.com Studio, where they appear in the Assets section of the Overview tab.
- **Create a Page Template** (Site administrator or designer)—Once you've decided on the layout, the quickest and most effective method is to use page templates to build the basic layout and then base your site pages on it. Try to keep the design of your main page template simple to make it easier to modify in the future. For more complicated site designs, such as the example graphic, you can use the main page template as the basis for a child template to achieve maximum flexibility. When you create your page template, you can choose from predesigned layouts that include headers, footers, and columns, or you can create a blank page template.
- **Lay Out the Page** (Site administrator or designer)—After you create the page template, you can modify the layout further to match the design of your site.
- **Create the Site Pages** (Site administrator or designer)—Using the template as a base, you can quickly create the site pages, which automatically inherit all the elements of the page template. Or if you need a standalone page that doesn't follow the site's overall design, you can create a blank page instead.
- **Add Features and Page Elements** (Site administrator or designer)—Use Site.com's prebuilt page elements to add features such as navigation menus, images, and data services, and include content blocks that contributors can edit. And add interactive, animated effects using events and actions.
- **Make Your Website Look Good** (Site administrator or designer)—Take advantage of cascading style sheets (CSS) to develop the look and feel of your website. If you're not completely up to speed with CSS, the Style pane provides an easy, visual way to create and manage styles. Or if you're a CSS expert who likes to get straight into the code, you can hand-code the site's style sheets.
- **Add and Edit Content** (Contributor)—At this stage, if you're a contributor, the site is usually ready for you to add and edit content such as text, images, videos, and hyperlinks. And as you work, you can upload any images or files you need.
- **Review and Test the Site** (Contributor, designer, or site administrator)—Testing the changes to the pages of your site happens throughout the development cycle. As a contributor, designer, or site administrator, you should always preview your changes to ensure they display as expected in a browser. And if you're a site administrator or designer, you can send a preview link to the site's reviewers so they can review the finished product before it goes live.

- **Publish the Site** (Site administrator only)—After testing is complete, you're ready to go live with your new site. Just set the site's domain information and then publish your changes to make your site live!

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

Using Site.com Studio as a Contributor

Setting Up the Contributor's Studio View

Site.com Tab Overview

If you can't see the Site.com tab, go to the Site.com app. It's available in the Force.com app menu in the Salesforce header. Then click the Site.com tab to view the list of your Site.com sites. From this page you can:

- Click **New** to create or import a site. Only users with the `Site.com Publisher User` field enabled on their user detail page can create and import sites.
- Filter the sites you see by selecting a predefined list from the drop-down list. My Sites shows the sites you can access and your role. All Sites shows all the sites in your organization even if you don't have access to some of them.
- Click **Edit** next to a site to open it in Site.com Studio.
- Click **Preview** next to a site to see how it looks when rendered in a browser window.
- Click ▼ next to a site to duplicate, export, or delete it. Only users with the `Site.com Publisher User` field enabled on their user detail page and the role of site administrator or designer can duplicate, export, and delete sites. If a site has been published, you can't delete it until you take it offline.
- See the status of your site.
  - In Development—The site has never been published.
  - Published—The site has been published at least once.
- Click the title of any column to sort your site list. By default, sites are sorted by name.

Note: You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

Using Site.com Studio as a Contributor

Planning and Implementing a Site.com Website

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To create or import Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

Using Site.com Studio as a Site Administrator or Designer

Site.com Studio provides a dedicated site-building environment for site administrators and designers. Using the many features available, you can:

- Create page templates to base your site pages on.
- Create site pages.
- Import assets, such as images and files.
- Edit the site's style sheet or create new style sheets.
- View and edit a page or template.
- Add page elements to your site pages to provide features and functionality.
- Use data services to connect to Salesforce objects to retrieve and display, or to submit data.
- Create custom widgets that you and other users can reuse throughout the site.
- Create a multilingual site that lets site visitors choose their preferred language.
- Create events to add interactive and animated effects to your website.
- Add IP restrictions to control site visitors' access to the pages, page templates, folders, and assets in your site.
- Add URL redirects to inform users and search engines if site content has moved.
- Create folders to organize your site content.
- Preview your site or generate an anonymous preview link to send to other users.
- Manage the domain information for your site.
- Publish your recent changes to the live site.
- Duplicate, import, and export sites.

Note:

- Designers can't manage domains or publish content.
- You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Understanding the Site Administrator and Designer's Overview Tab
Planning and Implementing a Site.com Website
Setting Up the Contributor's Studio View
Site.com Tab Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage sites Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Using Site.com Studio as a Contributor

Site.com Studio provides a dedicated content-editing environment for contributors, where you can:

- Open a page to edit it.
- Create site pages, if your site administrator or designer has enabled page creation.
- Edit the page text.
- Add images and hyperlinks to pages.
- Add page elements to pages.
- Import assets, such as images and files.
- Preview the site in a browser window.

SEE ALSO:

Understanding the Contributors's Overview Tab
Planning and Implementing a Site.com Website
Site.com Tab Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

Understanding the Site Administrator and Designer's Overview Tab

As a site administrator or designer, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and manage the site's components and configure the site's properties.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

To manage user roles:

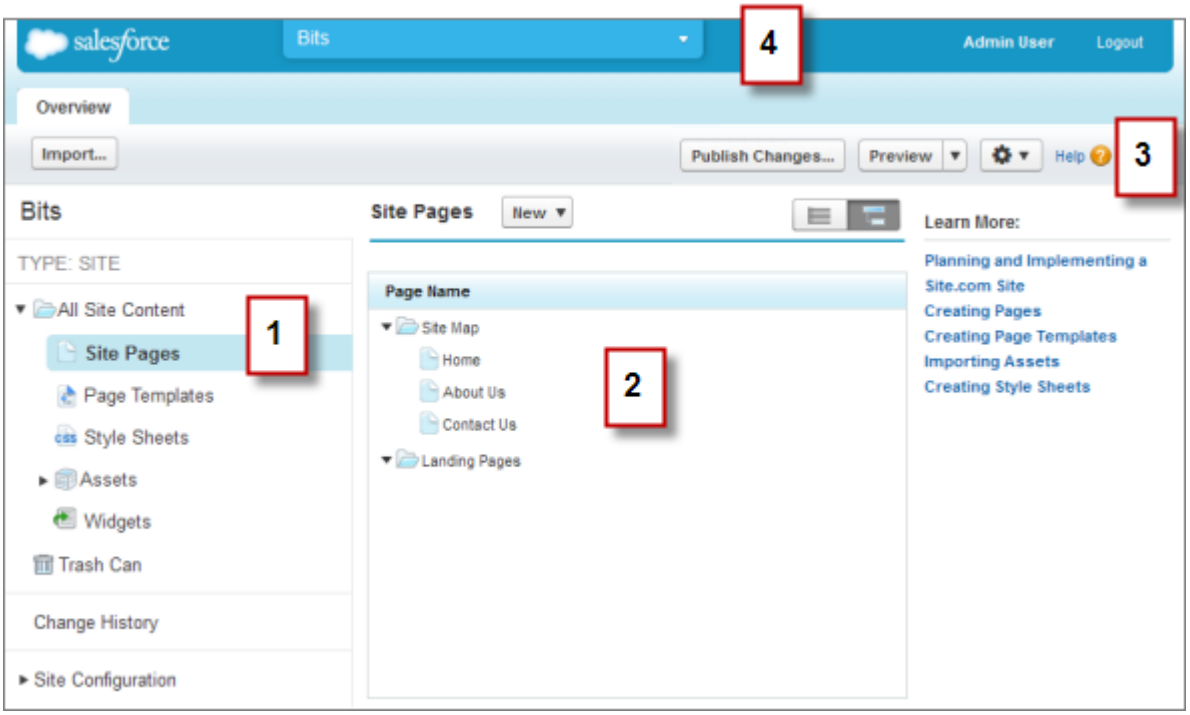
- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

  OR

  "Manage Users"



- Select a view (1) on the Overview tab to view its contents (2).
  - All Site Content—Create folders to organize your site content. In this view, you can also create pages, templates, and style sheets, and import assets.
  - Site Pages—Create site pages, open and edit pages, access page options, create site map links, and organize the site map. You can also switch between the default site map view ( ) and the list view ( ).
  - Page Templates—Create page templates to base your site pages on, open and edit existing templates, and access template options.
  - Style Sheets—Edit the site's style sheet or create new style sheets.
  - Assets—Import and manage assets, such as images and files.
  - Widgets—Build custom widgets that can you and your team can reuse throughout the site.
  - Trash Can—Retrieve deleted items. When you delete a page, template, style sheet, or asset, it goes into the trash can. Deleted items remain in the trash can indefinitely. Retrieved items are restored to their original location. If the original location no longer exists, they are restored to the top-level root directory.
  - Change History—View information about recently published files.
  - Site Configuration—Configure site properties, add IP restrictions, create URL redirects, manage domain information, manage user roles, and add and manage site languages.
- Use the toolbar (3) to:
  - Import assets, such as images and files.
  - Publish recent changes.
  - Preview your site or generate an anonymous preview link to send to other users.
  - Duplicate or export the site, overwrite the site with a version from sandbox, or create a new site ( ).
- Use the site's pull-down menu (4) to:

- Open recently accessed sites.
- View Site.com Studio as your contributors see it to ensure that you set up the view correctly.
- Exit Site.com Studio and return to Salesforce.
- Create a new site.
- Duplicate the site.

Note: You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

Planning and Implementing a Site.com Website

Site.com Overview

Understanding the Contributors's Overview Tab

As a contributor, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and edit the site's pages and content, and import images and files.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

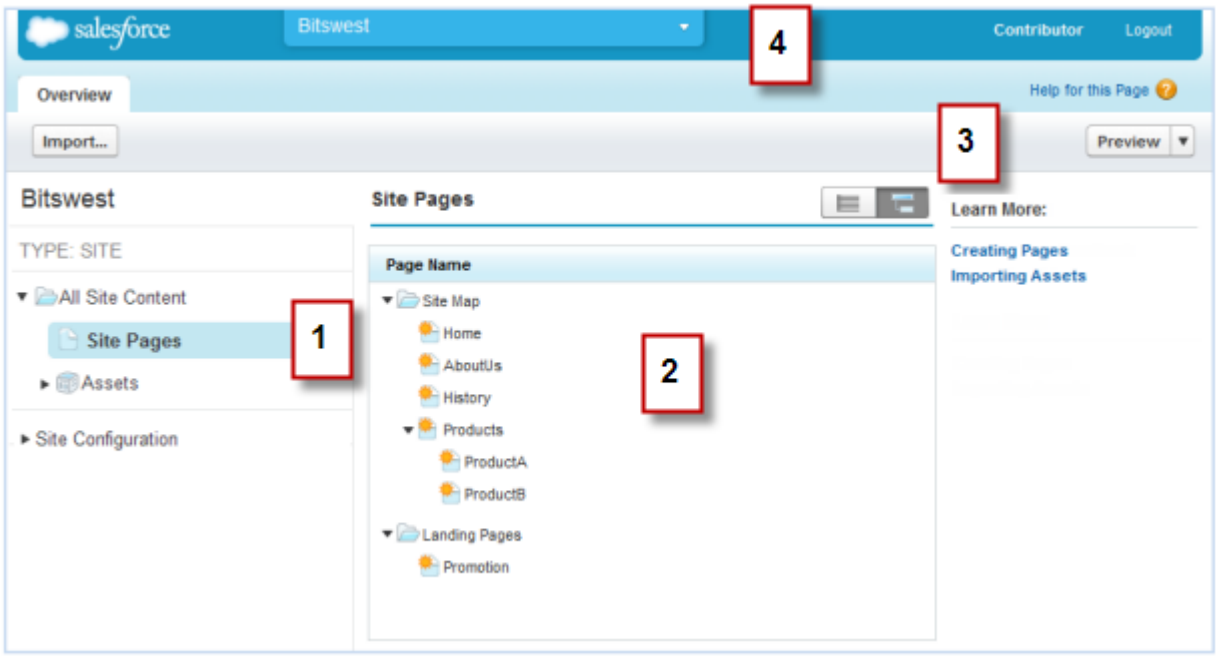
USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level



- Select a view (1) on the Overview tab to view its contents (2).
  - All Site Content—View all of the site's pages, images, and files.
  - Site Pages—View and edit pages or create site pages, if available.
  - Assets—Import assets, such as images and files.
  - Site Configuration—Manage user roles in the site. This is available only if you have the "Manage Users" perm.
- Use the toolbar (3) to:
  - Import assets, such as images and files.
  - Preview the site in a browser window.
- Use the site's pull-down menu (4) to:
  - Open recently accessed sites.
  - Exit Site.com Studio and return to Salesforce.

SEE ALSO:

Using Site.com Studio as a Contributor

Site.com Overview

Using Site.com with Communities

Use Community Builder (Site.com) Without Templates

Each community has one associated Site.com site that lets you add custom, branded pages to your community. By default, Site.com pages are publicly available and don't require login, but you can also create private pages that only community members can access.

For more detailed information about using Site.com, check out the online help.

Before You Begin

Communities users with the "Create and Set Up Communities" permission automatically have full site administrator access to a community's Site.com site. To let Communities users without the permission edit the site, you must purchase and assign either a Site.com Publisher or a Site.com Contributor feature license, and assign a user role at the site level.

See About Site.com User Roles on page 644.

Tips and Considerations

- Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. However, they don't appear in the User Roles section on the Overview tab of Site.com Studio.
- You can't create, delete, or duplicate community sites in Site.com.
- When working with data-bound components, such as data repeaters and forms, keep in mind that the objects listed may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.
- When adding forms to authenticated community pages in Site.com, set the current user for Salesforce objects that require the Owner ID field. Setting the current user (as opposed to the default guest user) lets you identify the authenticated user when the form is submitted. To set the current user for the Owner ID field, select the field in the form, click **Configure** under Field Properties in the Properties pane, select `Global Property` as the source, and select `Current userID` as the value.
- The home page, 404 page, login page, and self-registration page that you specify for Site.com Community sites in Site Configuration set the default pages for the Site.com Community site. These default URLs are used unless you specify different URLs in Community Management under **AdministrationPages** and **AdministrationLogin & Registration**. Community error pages are specified in Force.com Setup, under Error Pages.
- When your Site.com Community site is inactive, users are redirected to the Service Not Available page defined in Community Management under Pages.
- The contributor's view is not available by default for Site.com Community sites. However, you can use a Site.com Contributor license to grant contributor access to a specific user. See *About Feature Licenses* in the Site.com help for details. Alternatively, a user can preview the Site.com Community site as a contributor by appending `?iscontrib` to the site's URL. For example: `https://sitestudio.na1.force.com/?iscontrib`

SEE ALSO:

Use Site.com to Customize Your Community

Use Site.com to Customize Your Community

Communities users can use Site.com to build custom, branded pages for a community. There are many approaches to building custom pages for your community, but these are some of the typical stages involved:

- **Import Assets**—Collect the assets, images, and files you plan to use on your custom page. Import the assets into Site.com Studio, where they appear in the Assets section of the Overview tab.
- **Create Branded Pages**—The quickest and easiest way to create branded pages is to use the Community Template, which is automatically included with all Site.com community sites. When you create a new page based on the Community Template, the page includes all of the branded styles in your community, including the community's header and footer. If you want even more control over the look and feel of your community page, you can create your own page template, drag community headers and footers to it from the Widgets section of the Page Elements pane, and add other community styles.

  Note: As of Spring '15, the Community Template is no longer available for new communities. If you already have a Site.com community that's based on the Community Template, it will continue to work. For information on creating a community, see Create Communities.

- **Use Branded CommunityStyles**—Develop the look and feel of your custom pages by using the CommunityBranding style sheet, or by creating branded community styles in your own cascading style sheets (CSS). If you're not completely up to speed with CSS, the Style pane provides an easy, visual way to create and manage styles. Or if you're a CSS expert who likes to get straight into the code, you can hand-code community styles right in your own style sheets.
- **Create Public Pages**—Using the template as a base, you can quickly create pages, which automatically inherit all the elements of the page template. Or if you need a standalone page that doesn't follow the overall design, you can create a blank page instead.
- **Make Pages Private**—By default, any page you create in Site.com Studio is publicly available. However, you can make pages private so that only logged-in Communities users can access them.
- **Add Features, Page Elements, and Community Widgets**—Use Site.com's prebuilt page elements to add features such as navigation menus, images, content blocks, and community widgets. Retrieve data from your organization's objects and dynamically display it on your site pages using data repeaters and data tables. Alternatively, gather and submit data from visitors using forms.
- **Add and Edit Content**—At this stage, the page is usually ready for you to add and edit content such as text, images, videos, and hyperlinks. And as you work, you can upload any images or files you need.
- **Review and Test the Page**—Testing the changes to your page happens throughout the development cycle. You should always preview your changes to ensure they display as expected in a browser. You can also send a preview link to reviewers so they can review the finished product before it goes live.
- **Publish the Page**—After testing is complete, you're ready to make the page available to your community by publishing your changes.
- **Add Authenticated Pages to Your Community's Tab**—Now that the page is tested and published, if you're working with authenticated pages, the final step is to add the page to a tab in your community.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

- **Use Site.com in Sandbox**—Site.com is now available on sandbox. When you create a sandbox copy from a production organization, you can include your Site.com sites. You can also copy your sandbox site back to production using the overwrite feature.

SEE ALSO:

Use Community Builder (Site.com) Without Templates

Which Community Template Should I Use?

Community templates let you build a community that gives customers the same visual and functional experience on tablets, mobile devices, or their desktops.

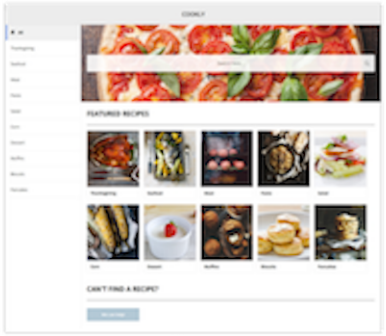
Your customers want an easy way to interact with your company whenever and wherever they are. A self-service community gives your customers an easy way access the information they need when they have questions. With communities built on the Koa and Kokua templates, customers can search for information using knowledge articles and contact support, if necessary. They can search for articles, ask questions, and get crowd-sourced answers from the community in communities built using the Customer Service (Napili) template. If they still can't find what they're looking for, they can contact support without having to log in. The self-service templates let you extend your community to various devices with a consistent user experience every time.

EDITIONS

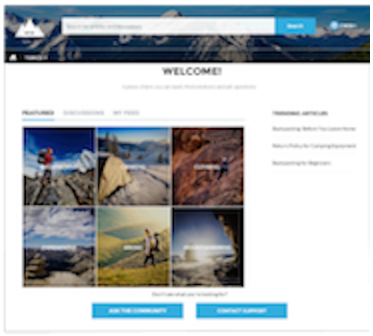
Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Koa



Kokua



Customer Service (Napili)

A wizard guides you through selecting a template and then walks you through the initial setup, including topic and data category setup.

- Kokua is a visually rich self-service template that presents users with knowledge articles organized by data categories. Users can also submit cases to get help from agents.
- Koa is a text-based self-service template that's optimized for mobile devices. It lets users search for and view articles by text representations of data categories, and contact support if they can't find what they're looking for.
- Customer Service (Napili) template lets your customers post questions to the community and search for and view articles. If they don't find what they're looking for, they can contact a support agent.

Community Builder makes it super easy to customize your community. Simply edit a few components to include information about your community, add images to extend your branding, and you're ready to go—without any coding! Also, if you want a more custom experience, you can create custom pages, add components to pages, use custom Lightning components, and expose more Salesforce objects.

Communities are automatically displayed in a logged-in user's profile language, so you don't need to create and manage a separate community for each language. With Community Builder, you can extend the language support for guest users in communities. If a guest

user's language isn't supported, they can choose one from a language picker on each community page. And because all languages are maintained within the community, you don't have to create and manage a separate community for each language.

SEE ALSO:

Configure and Customize Pages in Your Community

Implementation guide: Community Templates for Self-Service

Configure and Customize Pages in Your Community

Pages are the building blocks of your community. They control how your users see and interact with the community.

Each page contains a collection of components that are organized around a task or function—whether it's opening a case or searching for an article. The pages that comprise each of the preconfigured community templates are ready for you to use with very little configuration. Also, you can create pages or drag and drop other components on to existing pages as needed. If you create custom Lightning components, they appear in the list of available components on the left side of the page.

When you edit a page, you can set properties for how information appears, move components around, or delete components you don't require.

Note: You can delete inactive pages, but you can't rename pages.

For example, the Home page in the Customer Service (Napili) template contains components that quickly give community members an overview of the community content. These include components that let customers explore articles, read discussions, search for information, ask questions, and contact support. If you're creating a community to deflect cases and don't want to let customers file cases, you can remove the component for contacting support.

To edit a page, go to Community Builder

1. From the Page dropdown, select the page to edit.
2. Click  to open the Page Editor.
3. Select the component you want to edit.
4. In the property editor, edit any of the properties of the component.
5. To add a component to the page, simply drag it from the Lightning Components menu and drop it where you want it on the page.
6. To delete a component from the page, hover over the component and click  .
7. After you're done configuring the page, click **Publish**.

SEE ALSO:

Which Community Template Should I Use?

Implementation guide: Community Templates for Self-Service

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create, customize, or publish a community:

- "Create and Manage Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To enable Communities:

- "Customize Application"

To create article types and article actions:

- "Manage Salesforce Knowledge"

To create data categories:

- "Manage Data Categories"

Creating Branded Pages in Site.com

Create Branded Pages Overview

When you create a Community site, Salesforce automatically creates a new Site.com site and associates it with your community.

With Site.com Community sites you can:

- Use the branded Community template to create Site.com pages for your community.

  Note: As of Spring '15, the Community Template is no longer available for new communities. If you already have a Site.com community that's based on the Community Template, it will continue to work. For information on creating a community, see Create Communities.

- Use the `CommunityBranding` style sheet to style Site.com pages by using CSS.
- Create your own community CSS styles using a number of available `Network` namespace expressions.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Create Branded Pages from the Community Template

Site.com Community sites include a branded template that you can use to create new community site pages.

The styles for the `Community Template` come from the `CommunityBranding` style sheet, which is automatically included for all new Site.com Community sites.

To create branded pages from the `Community Template`:

1. On the Site.com Overview tab, hover over Site Pages and click **New**.
2. Type the new community page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Make sure `Community Template` is selected for the page template.
4. Click **Create**.

Note:

- Community branding options, such as headers, footers, and page colors, are set from the **Administration** > **Branding** section on the Community Management page.
- Empty community headers and footers, or headers that contain only images, won't work in Site.com. Be sure to specify customized HTML blocks for your community headers and footers if you're creating Site.com pages from the `Community Template`, or creating community headers and footers using `Network` namespace expressions.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

- Community headers and footers are available as widgets in Site.com community pages. To add a community header or footer to a blank page, drag it to the page from the Widgets section of the Page Elements pane.

SEE ALSO:

Getting Started With Communities
Create Branded Pages Overview
View the CommunityBranding Style Sheet
Site.com Page Templates Overview
Creating Site.com Page Templates

Apply Community Styles from the CommunityBranding Style Sheet

The `CommunityBranding` style sheet contains a set of CSS styles created from `Network` namespace expressions.

The `CommunityBranding` style sheet is attached to the `Community Template`, and is responsible for the template's branded look and feel. You can access the styles in the `CommunityBranding` style sheet and apply them directly to elements on any page.

To apply community styles using the `CommunityBranding` style sheet:

1. Make sure the `CommunityBranding` style sheet is attached to the Site.com page you want to brand. (See Creating and Using CSS Style Sheets on page 732.)

   Note: All Site.com pages based on the `Community Template` automatically have the `CommunityBranding` style sheet attached to them.

2. Select the element on the page you want to style.
3. Open the Style pane.
4. Select Class.
5. Start typing "brand".
   A list of all of the available styles in the `CommunityBranding` styles sheet appears.
6. Select the style you want to apply.

SEE ALSO:

Create Branded Pages Overview
Create Branded Pages from the Community Template
View the CommunityBranding Style Sheet
Creating and Using CSS Style Sheets

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Create Community Styles in a CSS Style Sheet

Branded Community styles are available in Site.com Community sites through `Network` namespace expressions.

You can access a full list of available `Network` namespace expressions to create new community styles in any CSS style sheet. When you add an expression to a CSS rule, Site.com "pulls in" the style as it's defined in the community, and displays it on your page.

To create community styles in a CSS style sheet:

1. Open an existing style sheet or create a new style sheet. (See Creating and Using CSS Style Sheets on page 732.)
2. Click **Edit Style Sheet Code**.
3. Add a new community style rule by using any of the available `Network` expressions. You can create both ID styles and class styles. For example:

```
#main_content_block {
  background-color: {!Network.primaryColor};
  color: {!Network.primaryComplementColor};
}
.secondary_content_blocks{
  background-color: {!Network.zeronaryColor};
  color: {!Network.zeronaryComplementColor};
}
```

4. Apply the new styles to elements on other pages.

Note: Remember, the style sheet that contains your community styles must be attached to the page containing your styled elements.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

Create Branded Pages Overview
Expressions Available for Community Branding
Creating and Using CSS Style Sheets

Expressions Available for Community Branding

You can use the `Network` namespace expressions listed on this page to create your own Community styles.

Community branding options, such as headers, footers, and page colors, are set from the **Administration** > **Branding** section on the Community Management page.

Note:

- Empty community headers and footers, or headers that contain only images, won't work in Site.com. Be sure to specify customized HTML blocks for your community headers and footers if you're creating Site.com pages from the `Community Template`, or creating community headers and footers using `Network` namespace expressions.
- Community headers and footers are available as widgets in Site.com community pages. To add a community header or footer to a blank page, drag it to the page from the Widgets section of the Page Elements pane.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Network Expression	Corresponding Community Branding Page Element
`{!Network.header}`	Custom content of the community header.
`{!Network.footer}`	Custom content of the community footer.
`{!Network.zeronaryColor}`	The background color for the community header.
`{!Network.zeronaryComplementColor}`	The font color used with `zeronaryColor`.
`{!Network.primaryColor}`	The color used for active tabs in the community.
`{!Network.primaryComplementColor}`	The font color used with `primaryColor`.
`{!Network.secondaryColor}`	The color used for the top border of lists and tables in the community.
`{!Network.tertiaryColor}`	The background color for section headers on edit and detail pages in the community.
`{!Network.tertiaryComplementColor}`	The font color used with `tertiaryColor`.
`{!Network.quaternaryColor}`	The background color for pages in the community.
`{!Network.quaternaryComplementColor}`	The font color used with `quaternaryColor`.

SEE ALSO:

- Getting Started With Communities
- Create Branded Pages Overview
- Create Community Styles in a CSS Style Sheet

View the CommunityBranding Style Sheet

The `CommunityBranding` style sheet contains a set of branded styles from your community.

Community branding options, such as headers, footers, and page colors, are set from the **Administration** > **Branding** section on the Community Management page.

To see the Community styles in the `CommunityBranding` style sheet, on the Site.com Overview tab, click Style Sheets, and click the `CommunityBranding` style sheet. The Community styles are listed on the left. To see the code for the style sheet, click **Edit Style Sheet Code**.

A total of fourteen Community class styles are provided. These are the default contents of the style sheet:

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

```
.brandZeronaryBgr {
 background-color: {!Network.zeronaryColor} !important;
}
.brandZeronaryFgr {
 color: {!Network.zeronaryComplementColor} !important;
}
.brandPrimaryBgr {
 background-color: {!Network.primaryColor} !important;
}
.brandPrimaryFgr {
 color: {!Network.primaryComplementColor} !important;
}
.brandPrimaryBrd2 {
 border-color: {!Network.primaryComplementColor} !important;
}
.brandPrimaryFgrBrdTop {
 border-top-color: {!Network.primaryComplementColor} !important;
}
.brandPrimaryBrd {
 border-top-color: {!Network.primaryColor} !important;
}
.brandSecondaryBrd {
 border-color: {!Network.secondaryColor} !important;
}
.brandSecondaryBgr {
 background-color: {!Network.secondaryColor} !important;
}
.brandTertiaryFgr {
 color: {!Network.tertiaryComplementColor} !important;
}
.brandTertiaryBgr {
 background-color: {!Network.tertiaryColor} !important;
 color: {!Network.tertiaryComplementColor} !important;
 background-image: none !important;
}
.brandTertiaryBrd {
 border-top-color: {!Network.tertiaryColor} !important;
}
.brandQuaternaryFgr {
 color: {!Network.quaternaryComplementColor} !important;
```

```
}
.brandQuaternaryBgr {
 background-color: {!Network.quaternaryColor} !important;
}
```

SEE ALSO:

- Getting Started With Communities
- Create Branded Pages Overview
- Create Branded Pages from the Community Template

Adding Authenticated Pages

Site.com Authorization Overview

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

As part of your site design, you might want to control what content is public and private to your site visitors. New sites are initially set so that all site resources, such as folders and pages, are public. You can change the default setting from the Authorization view found under Site Configuration.

The global site authorization options are:

- No Authorization (default)—All resources are public.
- Requires Authorization—All resources are private.
- Custom—All resources are public by default, but can be made private.

The No Authorization and Requires Authorization options let you quickly make your site either all public or all private. But, if you want to control access to individual pages, folders, and other resources, use the Custom option. Selecting Custom enables a `Requires Authorization` checkbox on the Actions menu for all resources throughout the site. You can define authorization at the site, folder, page, and individual resource level. As you mark items for authorization, a lock icon appears on them. After a resource, like a page, is marked as private, users who aren't logged into Salesforce are asked to log in when they try to access it.

Resources can inherit their privacy setting from folders. For example, when a resource, such as a site folder, is marked for authorization, anything placed in that folder inherits the folder's authorization setting and becomes private. If you drag that resource into a public folder, it becomes public again. But, if you explicitly mark a resource as private using the Actions menu, and then drag it into a public folder, it still remains private because the privacy setting at the resource level dominates.

When you use the Custom option, an authorization table appears in the Authorization view that lets you manage your private resources/items marked as private. You can remove authorization from a resource by either deleting it from the authorization table, or by deselecting the Requires Authorization box on the item itself.

SEE ALSO:

- Setting Authorization on Your Site
- Setting Custom Authorization
- Removing Site.com Authorization

Setting Authorization on Your Site

As part of your site design, you might want to control what content is public and private to your site visitors. New sites are initially set so that all site resources, such as folders and pages, are public. You can change the default setting from the Authorization view found under Site Configuration.

The global site authorization options are:

- No Authorization (default)—All resources are public.
- Requires Authorization—All resources are private.
- Custom—All resources are public by default, but can be made private.

1. Open your site for editing.
2. Click **Site Configuration** > **Authorization**.
3. Select an authorization option.

SEE ALSO:

Site.com Authorization Overview
Setting Custom Authorization
Removing Site.com Authorization

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To manage authorization:

- You must be an administrative user on the site

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Setting Custom Authorization

When you select Custom authorization, you get a great deal of flexibility in controlling access to your site. Not only can you control who has access to top level resources, like folders and pages, but you can also set access at the individual resource level.

Using Custom authorization at the folder level is a great way to make a large number of resources private without having to mark them individually. Let's say you periodically run sale offers for your paid users. If you drag all the sale pages into a special folder you mark for authorization, they instantly inherit the folder's setting. Users will need to log in to access them. Plus, if you decide to make one of the sale pages available to everyone, you can simply drag it back into a public folder, or to the root of the All Site Content area.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To manage authorization:

- You must be an administrative user on the site

1. Open you site for editing.
2. Click **Site Configuration** > **Authorization**.
3. Select **Custom**.
4. Click **All Site Content**.
5. Create a folder to hold the private pages if it doesn't already exist.
6. From the folder's Actions menu, select **Requires Authorization**. You'll see the lock appear on the folder. It is now private.
7. Drag any pages you want to make private into the folder. A lock appears on them too.

Example: Let's take another example. If you have a page that you'd like to keep private no matter where it resides, you can set its authorization using the Actions menu. After you set it at the individual resource level, it remains private even if you drag it into a folder that isn't set to private. In other words, an resource marked private is always private until you deselect **Requires Authorization** on the Actions menu.

If you check the Authorization page, you'll see all folders and resources marked private are listed in the authorization table where you can view and delete them.

SEE ALSO:

Site.com Authorization Overview
Setting Authorization on Your Site
Removing Site.com Authorization

Removing Site.com Authorization

You can remove authorization for a resource by either deleting it from the authorization table under Site Configuration, or by deselecting **Requires Authorization** from the menu.

1. Open your site for editing.
2. Click **Site Configuration** > **Authorization**.
3. From the authorization table, click **Delete** next to the item you want to remove. Alternatively, navigate to the All Site Content view. Select the resource. From the Actions menu, deselect **Requires Authorization**.

Example: If a resource is explicitly marked as private using the Actions menu, then you must remove authorization from it using the Actions menu. For example, if a page marked private is dragged into a folder that's public, it remains private. Likewise, if you drag it into a folder that's already private, and remove the authorization on that folder, the page will still be private.

SEE ALSO:

- Site.com Authorization Overview
- Setting Authorization on Your Site
- Setting Custom Authorization

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To manage authorization:

- You must be an administrative user on the site

Displaying Current Community User Information in Site.com

Display Current Community User Information

Site.com designers creating authenticated pages for a community site can display the current user's information by accessing `CurrentUser` namespace expressions.

1. Open the page on which you want to display the current community user's information.
2. From the Page Elements pane, drag a **Content Block** or **Custom Code** page element onto the page.
3. Type `{!CurrentUser.` and the value that you want to display.
   For example, `{!CurrentUser.firstName}`.

   Check the list of available expressions for displaying current user information.
4. Add any additional text you require.
   For example, `Welcome back {!CurrentUser.firstName}!`.
5. If you're in a Content Block, click **Save**. If you're in a Custom Code element, click **Save and Close**.

Note: If an unauthenticated user views a page that contains `CurrentUser` expressions, the current user information does not appear. For example, if an unauthenticated user viewed a page that contained the above example, the user would see "Welcome back !" as the welcome message.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Expressions Available for Displaying Current User Information

Use these `CurrentUser` namespace expressions to display authenticated user information on a Site.com community page.

CurrentUser Expression	Displays
`{!CurrentUser.name}`	Combined first and last name of the user, as displayed on the user detail page.
`{!CurrentUser.firstName}`	First name of the user, as displayed on the user edit page.
`{!CurrentUser.lastName}`	Last name of the user, as displayed on the user edit page.
`{!CurrentUser.userName}`	Administrative field that defines the user's login.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

CurrentUser Expression	Displays
`{!CurrentUser.email}`	Email address of the user.
`{!CurrentUser.communityNickname}`	Name used to identify the user in a community.

Determine the URL of a Site.com Page

After you create a Site.com page, you can determine the page's URL to:

- Provide your users with a URL that lets them access a public page directly.
- Create a link to the page from other pages, including Force.com Sites and Visualforce pages.
- Make it the home page for your community using a URL redirect in Force.com Sites.
- Add a private page to a web tab in your community.

1. To determine the correct URL for the page:
   - From the Create Community wizard, click **Customize**.
   - If you navigated away from the Create Community wizard, click **Customize** > **Communities** > **All Communities**, then click the **Manage** button next to the community name.
2. Click **AdministrationSettings**.
3. Copy the URL displayed on the page and paste it into a text editor.
4. To create a URL that points to:
   - The Site.com site's home page, append `/s/` to the URL. For example, `https://mydomain.force.com/mycommunity/s/`.
   - A specific Site.com page, append `/s/<page_name>`, where `<page_name>` is the name of the Site.com page. For example, `https://mydomain.force.com/mycommunity/s/promotion`.

   The URL is case-sensitive and "s" must be lowercase.

SEE ALSO:

Add Authenticated Site.com Pages to Community Tabs

Force.com Sites URL Redirects

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Add Authenticated Site.com Pages to Community Tabs

After you create a private Site.com page, you can add the page to a tab in your community.

In this case, you need to create a Web tab that points to your Site.com page.

1. In the Properties pane for your page, select `Show Salesforce Header`.
   Selecting this option ensures that you see tabs in your community.
2. Enter the tab name as it should appear on the tab in your community.
   The web tab you create must have the same name.
3. Determine the correct URL for the page.
   The URL must be in the following format `https://mydomain.force.com/mycommunity/s/<pagename>`, where *pagename* matches the name of your page.
4. From Setup, enter *Tabs* in the `Quick Find` box, then select **Tabs**.
5. In Web Tabs, click **New** and enter the name of the tab as it appears in the Tab Name field in your page properties.
6. On the Step 3 screen, paste the URL you created in the Button or Link URL text box.
7. Return to the Create Community wizard and add the new tab to your community.

To preview the private page in your community, you must publish your Site.com site.

Note: You can't publish your site from sandbox.

SEE ALSO:

Create Web Tabs

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Add Chatter News or Group Feeds to Community Site.com Pages

Use the Chatter News Feed to display a Chatter feed on your site pages, or display the feeds of a particular group using the Chatter Group Feed.

1. Drag the **News Feed** or **Group Feed** from the Widgets section of the Page Elements pane onto the page.
   When you add a widget to a page, it creates a copy or *instance* of the widget. You can't edit the content of a widget, but you can edit the properties.
2. If you're adding a group feed, enter the Group ID in the Properties pane.
   The Group ID determines which group feed is displayed on your page. You can include more than one group feed on a page if you want to show the feeds for multiple groups.
3. Preview the page to test the feed, or use Live Mode to see how the feed renders in different mobile devices.

Consider the following limitations when using a news or group feed in your community Site.com sites:

- Chatter news and group feeds only appear if a user is logged in to the community. They don't appear to guest users or in anonymous preview mode.
- Chatter news and group feeds may not render appropriately on pages less than 700px wide. We recommend a minimum page width of 700px to view full content. We also recommend using a white background.
- Chatter news and group feeds only inherit some page branding elements.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Improve Performance with HTML Page Caching for Communities in Site.com

HTML caching lets you improve the performance and page rendering of your community's Site.com site by controlling how often the generated markup of the page is reloaded.

Lets say 100 people visit the page at the same time. Without caching, the page makes 100 separate requests for the same markup, slowing performance considerably. However, with caching enabled, the page markup is requested and retrieved only once—the first time someone visits the page. Any subsequent page requests during a set time period are returned from the cache. When the specified time period expires, the cache is refreshed.

Note: The caching duration applies only to community pages that are accessed by guest users. When a user logs in to access the page, caching is disabled.

1. In Site.com Studio, open the page.
2. In the `Cache Duration (Minutes)` field of the Cache section of the Properties tab, specify the length of time to cache the page.

   By default, the caching duration of a page is set to 30 minutes.

   To disable caching, set the page's caching duration to 0.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage a community's custom pages:

- "Create and Set Up Communities"

  OR

  `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating and Managing Sites

Creating a Site.com Site

To get started with Site.com, create a new blank site:

1. On the Site.com tab in the Site.com app, click **New**. Alternatively, in Site.com Studio, click **Create a New Site** in the site's drop-down menu.
2. Click **Create a Blank Website**.
3. Enter the site name.
4. Click **Create**. Your new website opens in Site.com Studio, where you can create page templates and site pages, and add functionality to it.

Note: You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Creating Site.com Page Templates

Creating Site.com Pages

Editing Site.com Pages as a Designer or Site Administrator

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To create or import Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

Deleting a Site.com Site

You can delete any site that isn't published. If the site is published, you must first unpublish it before you can delete it. See Taking a Site Offline on page 813.

1. On the Site.com tab in the Site.com app, select the site and click ▾ > **Delete**.
2. Click **OK**.

Note: You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Exporting a Site.com Site

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Duplicating a Site.com Site

To create a copy of a site:

1. On the Site.com tab in the Site.com app, select the site and click > **Duplicate**. Alternatively, on the Overview tab in Site.com Studio, click  > **Duplicate This Site**.
2. Enter a name for the site.
3. Click **Create**.

Note:

- If you're creating a copy of a site that uses data services, you must set the data access permissions in the new site's guest user profile.
- You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Creating a Site.com Site

Exporting a Site.com Site

Importing a Site.com Site

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Exporting a Site.com Site

You can export your site from Site.com to your hard drive. The site is exported in a packaged format with a .site extension, which you can import into another Salesforce organization. The maximum site size you can import is 2 GB.

1. On the Site.com tab in the Site.com application, select the site and click > **Export**.

   Alternatively, on the Overview tab in Site.com Studio, click > **Export This Site**.
2. If the site is:
   - Smaller than 100 MB, select a location to save the exported .site file on your hard drive and click **Save**.
   - Larger than 100 MB, you'll receive an email when the export process has completed. Click the link in the email to download the exported .site file.

Note:

- Exporting a site doesn't remove it from the current organization.

SEE ALSO:

Creating a Site.com Site

Importing a Site.com Site

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Importing a Site.com Site

You can import an exported Site.com site into your organization. The maximum site size you can import is 2 GB. When you import a site, you are given the site administrator role in the site.

1. On the Site.com tab in the Site.com app, click **New**. Alternatively, in Site.com Studio, click **Create a New Site** in the site's drop-down menu.
2. Select **Import a Site or Template**.
3. Enter the site name.
4. Click **Browse** to locate the exported site on your hard drive. Exported sites have a .site extension.
5. Click **Create**.

Note:

- If you're importing a site that uses data services, you must set the data access permissions in the imported site's guest user profile. Additionally, any data repeaters, data tables, data functions, or forms may need to be reconfigured.
- You can't create, delete, or duplicate community sites in Site.com.

SEE ALSO:

Creating a Site.com Site

Exporting a Site.com Site

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To create or import Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

Configuring Site Properties

Set properties for the site, such as the home page, site name, and error page, and create an anonymous preview URL that allows other users to review the site before it goes live. The URL is always valid (unless you disable it) and shows the latest work in progress. It's only available to the people you send it to, and can't be found by search engines.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

1. On the Overview tab, click **Site Configuration**.
2. Click **Edit**.
3. In the Site Configuration view, you can:
   - Replace the name in the `Site Name` field to rename the site.
   - See the `Developer Name` of the site. This read-only field may differ from the `Site Name`. The `Developer Name` is used by the Metadata API.
   - Select **Enable Anonymous Preview** to create a URL that allows other users to preview the site before it goes live. (Click the **View Anonymous Preview** option that appears in the **Preview** menu to access the preview URL, which you can copy and send to other users to review and test your changes.) **Enable Anonymous Preview** is also available from the **Preview** menu on the Overview tab.
   - Access the site's guest user profile.
   - Set the clickjack protection level.
   - Select the home page for your website in the Home Page drop-down list.
   - *Site.com Community sites only:*
     - Select the login page for your Site.com Community site in the Login Page drop-down list.
     - Select the page that you've set up for Site.com Community site users who don't have accounts yet from the Registration Page drop-down list.
     - Select the Forgot Password page you've set up for your community using Site.com.
   - Select a user-friendly error page in the 404 Page drop-down list to display when a page can't be found. It's a good idea to create a user-friendly error page to assist site visitors if they encounter a broken link.
4. Click **Save**.

Note:

- The home page, 404 page, login page, and self-registration page that you specify for Site.com Community sites in Site Configuration set the default pages for the Site.com Community site. These default URLs are used unless you specify different URLs in Community Management under **AdministrationPages** and **AdministrationLogin & Registration**. Community error pages are specified in Force.com Setup, under Error Pages.
- When your Site.com Community site is inactive, users are redirected to the Service Not Available page defined in Community Management under Pages.

SEE ALSO:

Creating a Site.com Site

Using Site.com Studio as a Site Administrator or Designer

Enabling Clickjack Protection

Enabling Clickjack Protection

Clickjacking is a type of attack that tries to trick a user into clicking something, maybe a button or link, because they perceive they are clicking something safe. Instead, the button or link performs malicious actions on your site leading to data intrusion, unauthorized emails, changed credentials, or other site-specific actions.

Hidden iframes that load your site's pages can be placed maliciously by an unrelated page that entices the user to click a button or link that appears below the hidden iframe. With clickjack protection, you can configure whether your browser allows frames or iframes over your site pages. The default clickjack level for Site.com is set to Allow framing by the same origin only.

You can set the clickjack protection for a site to one of these levels:

- Allow framing by any page (no protection)
- Allow framing by the same origin only (recommended)
- Don't allow framing by any page (most protection)

Note: Same-origin framing allows the site's page to be framed only by pages on the same domain name and protocol security.

1. On the Overview tab, click **Site Configuration**.
2. Click **Edit**.
3. Select the desired level of clickjack protection.

   Note: Salesforce Communities have two clickjack protection parts—one for the Force.com Communities site which is set from the Force.com site detail page and another for the Site.com Communities site which is set from the Site.com configuration page. It's recommended that both are set to the same value.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

Configuring Site Properties

Creating and Editing Force.com Sites

Site.com Versioning Overview

Each time you publish your site, it's tracked as a version. You can restore your site back to one of the previously published versions. You can't select individual components when restoring; you must restore the complete site.

When working in Site.com Studio, you're always working on an unpublished version of your site. It's your *working copy*. When you restore a version, you overwrite your working copy, not your live site. You must publish the restored version before you see the change on your live site.

In Site.com Studio, you'll find your site versions in the Change History view on the Overview tab. Not all items in a site are reverted when you restore a version. Some things, like user roles, remain unchanged even when you restore from a previous version. Everything in a site is under version control except:

- Site name
- Anonymous preview setting
- Guest User Profile settings

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

- Clickjack protection level
- Domains and path prefixes
- User role settings

SEE ALSO:

Restoring to a Previous Site Version

Restoring to a Previous Site Version

The Change History list in Site.com Studio tracks all published versions of your site. You can select any previously published version and restore it. You can only restore an entire site, not parts of a site.

When working in Site.com Studio, you're always working on an unpublished version of your site. It's your *working copy*. When you restore a version, you overwrite your working copy, not your live site. You must publish the restored version before you see the change on your live site.

Warning: You can't restore the working copy of your site after you revert to a previous version. Therefore, it's a good idea to back up the working copy of the site before reverting to ensure you don't lose any unpublished changes.

To revert to a previously published site version:

1. Select the Overview tab.
2. From the Change History view, select the version you want to restore.
3. Click > **Restore Version**.
4. Click **OK** at the confirmation message.

After you restored your working site to a previous version, you can continue to make additional changes until you're ready to publish the site.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating URL Redirects in Site.com

If you move or reorganize pages in your site, search engines may have trouble finding the new page locations. To avoid this, set up URL redirects to inform users and search engines that site content has moved.

You can use URL redirects to:

- Maintain search engine ranking. For example, if you change a page's name from "Gadgets" to "Widgets," creating a redirect rule from `/Gadgets` to `/Widgets` lets you restructure the site without affecting your page ranking.
- Make URLs more readable and memorable. For example, site visitors will find long or numeric page names, such as `/widget65AD890004ab9923`, difficult to remember. Instead, you can provide them with a short, friendly URL, such as `/widget`, and create an alias that redirects to the correct page when the user uses the short URL alias.
- Assist with migration from another system to Site.com if you're still using the same domain. For example, if your old site ran on PHP, you can create a redirection rule from an old page, such as `/index.php`, to a new page in Site.com, such as `/myNewPage`.

To assign a redirect to a site page:

1. On the Overview tab, click **Site Configuration** > **URL Redirects**.
2. Click **Create a Redirect**.
3. Specify the `Redirect type`:

Option	Description
Permanent (301)	Select this option if you want users and search engines to update the URL in their systems when visiting the page. Users visiting a page redirected with this type are sent seamlessly to the new page. Using a permanent redirect ensures that your URLs retain their search engine popularity ratings, and that search engines index the new page and remove the obsolete source URL from their indexes.
Temporary (302)	Select this option if you want users and search engines to keep using the original URL for the page. Search engines interpret a 302 redirect as one that could change again at any time, and though they index and serve up the content on the new target page, they also keep the source URL in their indexes.
Alias	Select this option if you don't want the URL to change in the user's browser, but you want to redirect to a different page. Search engines won't be aware of the change or update their records.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Option	Description
	Alias redirects only work when you redirect from one Site.com page to another. You can't create an alias to an external address.

4. Specify the former page location in the `Redirect from` field.
   - The page location must be a relative URL.
   - The page can have any valid extension type, such as `.html` or `.php`, and can contain parameters. Parameter order is irrelevant.
   - The URL can't contain anchors, such as `/siteprefix/page.html`**`#target`**.
   - You can create just one redirection rule *from* a particular URL. If you create a new rule with the same Redirect From information, the old rule is overwritten.
5. Specify the new page location in the `Redirect to` field. This can be a relative URL or a fully-qualified URL with an `http://` or `https://` prefix. Unlike pages you're redirecting from, pages you're redirecting to can contain anchors.
6. To immediately enable the redirection rule, ensure `Active` is selected. To enable it at a later stage, deselect the property.
7. Click **Save**.

The URL Redirects section displays all URL redirection rules you've created for your site.

- Edit an assigned redirect rule by clicking > **Edit Redirect**.
- Delete a redirect rule by clicking > **Delete Redirect**.

Importing External Websites into Site.com

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

With Site.com Studio, you can import your existing website and recreate it automatically as a Site.com site. This saves you the time of having to re-code existing HTML pages.

Create a zipped file of your website and the content with the desired folder structure. Create a new Site.com site and then import your zipped file.

During the creation of the new Site.com site, all your HTML pages and assets are copied into the new site in the same locations they are in the zipped file. Here are some guidelines for importing assets.

- Some code in style sheets may not convert properly to the Site.com format. If this happens, you'll receive a warning message. You can continue to import the zipped file or stop the import. If you continue, the style sheet imports and you can then manually attach it to your pages.
- You can edit the `<head>` section of a page using Edit Head Markup in scripts section on the properties pane. This is useful for manually editing the HTML.
- The maximum file size you can import is 50 MB unless you import and *unzip* a .zip file. In that case, you can import a .zip file of up to 200 MB if you select **Unzip files** during the import process. If your site is larger than 200MB when zipped, you can create more than one zipped file and import them individually.
- Site.com attempts to format links correctly when importing a page. Links are checked in content blocks, custom code, and head script markup. You can check for any links that might still be broken on your page using the link checker. Open the page, select > **Find Broken Links**. A dialog displays showing any broken links. To fix the link, click **Edit**. The link opens in the HTML editor.

SEE ALSO:

Importing a Site.com Site

Copying and Overwriting a Site

Sometimes, you might want to copy and replace your current production Site.com or Site.com community site with another site.

You'll first need to export a copy of your site from your . Once you have an exported `.site` file, import that file into your production organization using the overwrite feature. Overwrite replaces your production site with the imported site. You can only overwrite sites that are the same type. For example, you can't overwrite a Site.com site with a Site.com community site.

1. Create your `.site` file using the export feature in Site.com Studio.
2. From Site.com Studio in your production site, click **Site Actions** > **Overwrite This Site**. Alternatively, if you have access to the Site.com tab in your organization, you can click next to the site name to find **Overwrite**.
3. Click **Browse** to find the .site file you exported.
4. Click **OK** at the overwrite warning.

Warning: You can't revert once you have overwritten a site.

Here are a few tips when using overwrite.

- You'll need to publish the site you overwrote before changes take effect.
- When copying a site, the production site is overwritten. So, be sure to backup your production site by exporting a `.site` file.
- Overwrite does not copy data changes. For example, if you created a custom object to use in a repeater in the site you are copying from, that repeater won't work in the production site until you create the same custom object.
- If there are assets that exist in the production site that do not exist in the site you are copying from, they are moved to the trash can during the overwrite process so you can restore them if needed.
- If your copied site has a different name than your production site, the production name is preserved and only the contents are changed. For example, if your site is called Site One and you overwrite your production site called Site Two, the production site is still called Site Two.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To overwrite sites:

- Site administrator or designer role assigned at the site level

SEE ALSO:

Exporting a Site.com Site

Using the Metadata API to Deploy a Site

Using the Metadata API to Deploy a Site

As a user, you can migrate a site from sandbox to production. You can use the Metadata API to create a deployable package for Salesforce Communities sites and Site.com sites.

There are several tools that you can use to create a package:

- Change sets (for Site.com and Site.com Communities sites only). The component type is called `Site.com`.
- Workbench, which works for creating all site types. The metadata type is called `SiteDotCom`.
- Force.com migration tool for Ant, which works for creating all site types. The metadata type is called `SiteDotCom`.

If using change sets, select Site.com from the list and follow the prompts to create your package. If using Workbench or Force.com as your tool, you must create a `package.xml` file. That file is submitted to the Metadata API to create a package.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Note: You can include a Guest User Profile in your `package.xml` file. If you do so, that Guest User Profile is linked to the site during deployment.

The packaging process generates a folder that contains a content file and a metadata xml file. The content file name is `[sitename].site`. The metadata `.xml` file name is `[sitename].site-meta.xml`.

If you deploy a package that doesn't include a `.site` file, an empty site is created. If the package contains a site file, and the organization already contains a site with the same name, the site is updated.

Note: There is a file size limitation when using the Metadata API to deploy a site from sandbox to production. The assets in the `.site` file can't be larger than 40 MB. The site gets created, but the assets show in the new site as broken. To fix the assets, export the assets from the sandbox environment separately and then import them into your new site.

For help with the Metadata API, see the Metadata API Developer Guide. You can find help for change sets in the online help and for the Force.com Migration Tool Guide at https://developer.salesforce.com/page/Migration_Tool_Guide.

SEE ALSO:

Sample Package.xml Files

Sample Package.xml Files

Here are some sample Site.com `package.xml` files.

Here is a sample `package.xml` file for a Site.com site.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>xyzsite</members>
        <name>SiteDotCom</name>
    </types>
    <version>30.0</version>
</Package>
```

Here is an example of a `package.xml` for a Salesforce Communities site. The package also includes a Guest User Profile.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>xyzsite</members>
        <name>CustomSite</name>
    </types>
    <types>
        <members>xyzsite</members>
        <name>Network</name>
```

```
    </types>
    <types>
        <members>xyzsite1</members>
        <name>SiteDotCom</name>
    </types>
    <types>
        <members>xyzsite Profile</members>
        <name>Profile</name>
    </types>
    <version>30.0</version>
</Package>
```

Managing Site Assets

Importing and Managing Assets

Contributors, publishers, and site administrators can import a variety of assets, such as images, HTML pages, and PDFs, to use in a website. You can import assets and files individually, or use a zipped file. When importing entire websites or large numbers of assets, it's easier to create a zipped file of the content with the desired folder structure. When importing the zipped file for a website, Site.com recreates your website and places everything in the same folder structure.

Note:

- The maximum file size you can import is 50 MB unless you import and *unzip* a .zip file. In that case, you can import a .zip file of up to 200 MB if you select **Unzip files** during the import process.

The quickest way to import one or more files is to:

1. Select the files from your computer and drag them directly onto the Studio interface. This is supported for Mozilla® Firefox® and Google® Chrome only. You can drag individual files, or a zipped file.
2. Depending on the types of files you're importing, a dialog box may appear that lets you:
   - Select **Unzip files** to extract the contents of a .zip file. If the .zip file includes folders, this structure is maintained in your site.
   - Select **Overwrite existing files** to replace a file that already exists in the site.
   - Select **Convert CSS files into style sheets**, if you're a site administrator or designer, to convert a CSS file into a style sheet that you can use to style your website.

     Note: If you import a .zip file that includes CSS files, and they fail to convert, they may not be valid. Try unchecking this option and then re-importing the .zip file.
   - Select **Convert HTML files into pages** to import HTML pages into your website. The structure of the HTML page is maintained in your site, but the HTML is not validated during import.

Alternatively, to import a single file:

1. Click **Import...**.
2. Click **Browse...** to locate the file.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

3. Select the file and click **Open**.
4. Depending on the type of file you're importing, you can:
   - Select **Unzip files** to extract the contents of a .zip file. If the .zip file includes folders, this structure is maintained in your site.
   - Select **Overwrite existing files** to replace a file that already exists in the site.
   - Select **Convert CSS files into style sheets**, if you're a site administrator or designer, to convert a CSS file into a style sheet that you can use to style your website.

     Note: If you import a .zip file that includes CSS files, and they fail to convert, they may not be valid. Try unchecking this option and then re-importing the .zip file.

   - Select **Convert HTML files into pages** to import HTML pages into your website. The structure of the HTML page is maintained in your site, but the HTML is not validated during import.
5. Click **Import**. A message appears indicating whether the file was imported successfully.
6. Click .

You can add images and videos to the text areas of your site pages, or create a hyperlink to any imported asset. If you're a site administrator or designer, you can add also add images directly to the page.

View the list of imported assets in the Assets view on the Overview tab. You can also access assets in the All Site Content view, which displays the folder hierarchy of your site.

- To view a thumbnail of an imported image, hover over it.
- To save an asset to your computer, hover over or select it and click > **Download**.
- To remove an asset from your site if you're a site administrator or designer, hover over or select it and click > **Delete**. If the asset is being used in your site, you see a confirmation message with a list of locations where that asset is in use.
- To rename an asset on your site if you're a site administrator or designer, hover over the asset and click > **Rename**.

SEE ALSO:

Adding Site.com Page Elements

Using Site.com Studio as a Site Administrator or Designer

Using Site.com Studio as a Contributor

Exporting Assets

Exporting Assets

Designers and site administrators can export all site assets separately from the `.site` file. You can export assets together or individually.

When you export all assets, the assets are exported to a zipped file that is named after the site name—for example, *`sitename`*`-Assets.zip`.

Export your assets from the Overview tab by clicking  > **Export Site Assets**.

To export a single asset, hover over the asset and select **Download** from the Actions menu.

Note: There is a file size limitation when using the Metadata API to deploy a site from sandbox to production. The assets in the `.site` file can't be larger than 40 MB. The site gets created, but the assets show in the new site as broken. To fix the assets, export the assets from the sandbox environment separately and then import them into your new site.

SEE ALSO:

Importing and Managing Assets

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating and Managing Folders

As a site administrator or designer, you can create folders to manage your pages, style sheets, templates, and assets.

To create new folders:

1. In the All Site Content view on the Overview tab, click **New Folder**.
2. Type in the folder name.
3. Click **Create**.

Folders are created at the top level of the folder tree. Once created, you can drag them anywhere in the tree structure. Likewise, you can drag and drop files into the folders you create. To rename, delete, and create sub-folders, right-click the folder or use the Actions menu ( ).

Note: The site map remains the same regardless of how you arrange folders in the All Site Content view.

SEE ALSO:

Understanding the Site Administrator and Designer's Overview Tab

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating and Managing Pages

Editing Site.com Pages as a Designer or Site Administrator

When working with page templates and site pages, you can add content, structure, and style, all in one place. Open a page or template on the Overview tab by double-clicking it or hovering over it and clicking > **Edit**. The page opens as a new tab.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

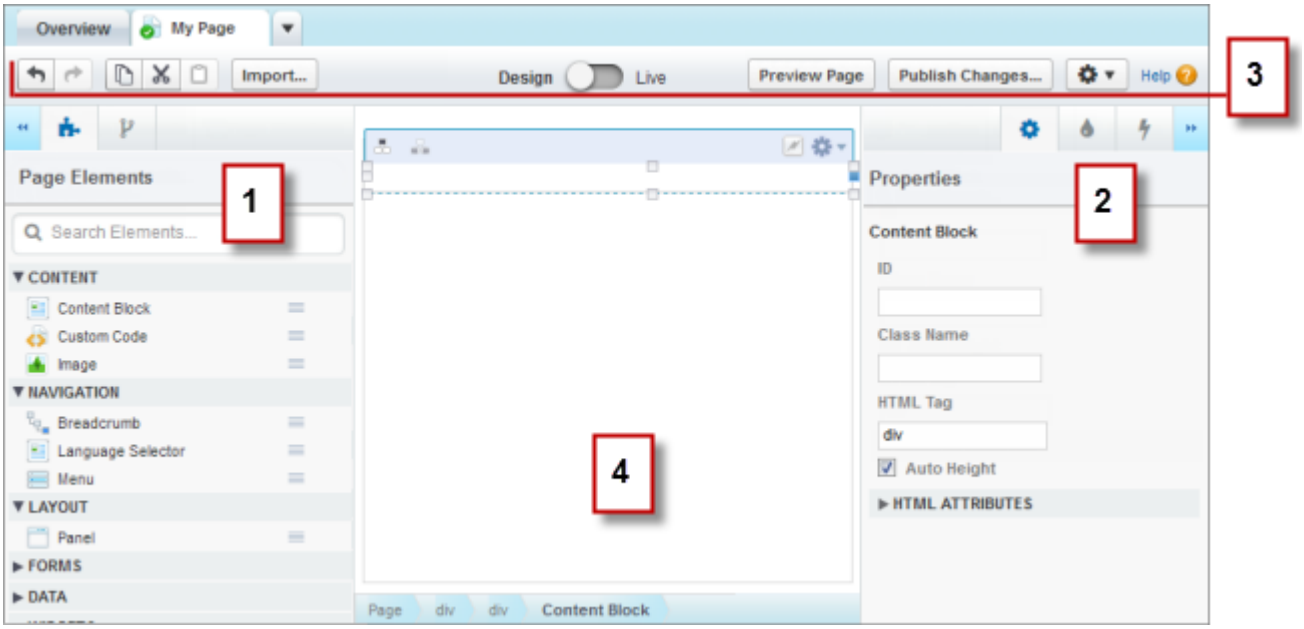
USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level



- Using the Page Elements and Page Structure panes (1), you can search for and add page elements to a page, and reorder the page element hierarchy.
- Using the Properties, Style, and Events panes (2), you can set properties, add style, and create events for a selected page or element.
- Using the toolbar (3), you can:
  - Undo and redo your actions.
  - Cut, copy, or paste page elements.
  - Import assets, such as images and files.
  - Preview pages to see how they'll appear when live on different devices.
  - Preview your site or generate an anonymous preview link to send to other users.
  - Publish your recent changes.
  - Access other page actions ( ), such as renaming or deleting the page.
- On the page canvas (4), you can lay out the page and select, edit, and move page elements.

Tip:

- Hide the side panes to increase the canvas size by clicking and . To reopen a pane, click its icon.
- As you edit a page, your changes are saved and the status icon ( ) on the page tab is updated automatically.
- If the site page or page template is based on another template, editable page elements are highlighted with a blue border on the page.

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

Site.com Page Templates Overview

Before you begin building the pages of your website, take some time to plan the pages you need, and in particular, which pages will have a similar layout. Once you've decided on the layout, the quickest method is to use a page template to build the basic layout.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

About Page Templates

A page template lets you define the layout and functionality of site pages in one location. By adding common page elements to the template and then basing site pages on it, you can achieve a consistent look and feel throughout your site. Page templates don't appear on your public site.

As the template creator, you specify which elements users can edit in pages based on the template. By default, a page element in a template is "locked," so users can't edit its contents in any template-based page unless you mark the page element as "editable." Conversely, when users edit an editable page element in a template-based page, their changes are specific to that page and don't affect your template.

For example, this main page template contains a non-editable header and navigation menu that are common to all the pages in the site (1). The main template also has an editable center panel (2) to house the page-specific content of each page that's based on it.

Note:

- Page templates must contain at least one editable page element. Otherwise, users can't edit site pages that are based on the template.
- Panels are ideal for adding editable areas to page templates.

You can use page templates to:

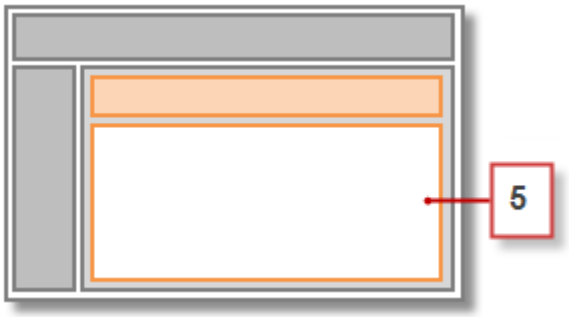
- Save time and effort by laying out the page structure and using it as a starting point when you create site pages. For example, you could design a template with a fixed header panel and side menu, and an editable center panel, to which you add page-specific page elements and content.
- Quickly make global updates to the layout or style of your website, as any changes you make to the template's design are reflected immediately in all the pages that use it.
- Control how other users (such as contributors or other site administrators and designers) can modify site pages. For example, you may allow contributors to edit specific content blocks only.
- Ensure your template design remains pixel-perfect. When users edit a page that's based on a template, their changes don't affect your template.
- Reuse common design elements by creating child templates.
- Allow contributors to create site pages that are based on the template.

About Child Templates

Child templates are a useful way to reuse common design elements for more complicated page layouts. For example, your website will probably have elements that are the same on every page in your site, such as a navigation menu. However, several pages may have elements that are common only to them, such as pages in a subsection of your site that include a subsection header. By using a child template, which is a template that's based on another template, you can reuse the main template design.

Using our main page template as a base, the child template inherits the non-editable header and navigation menu (1), and an editable center panel (2) where we add the non-editable subsection header (3). We also need to add a new editable center panel (4) because the center panel of the main template is editable only in pages directly based on the main template.

Now, any page based on the child template includes the non-editable main header, navigation menu, and subsection header, and an editable center panel (5) for that page's content.



Best Practices

- Plan your site structure and the layout of your pages. Taking the time to plan your website first saves time when you build your site.
- Identify which elements are common to all the pages of your site, such as navigation menus or headers, as these are the elements you can add to the page template.
- Use page templates wherever possible to promote content reuse and save time.
- Try to keep the design of your main page template as simple as possible to make it easier to modify in the future. For more complicated site designs, use child templates to achieve maximum flexibility.

SEE ALSO:

Creating Site.com Page Templates

Creating Site.com Pages

Setting Up the Contributor's Studio View

Identifying Which Template a Site.com Page Uses

Creating Site.com Page Templates

A page template lets you define the layout and functionality of site pages in one location. By adding common page elements to the template and then basing site pages on it, you can achieve a consistent look and feel throughout your site. And because a template-based page inherits the template's elements, you can make site-wide changes from one location.

You can create a page template from a layout, or if you've already created a template, you can use it as a base to create a child template, which lets you reuse the design of the main template.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Creating a Page Template from a Layout

To start from scratch with a completely blank template or use a basic page layout:

1. Hover over **Page Templates** on the Overview tab and click **New**, or click **New Page Template** in the Page Templates view.
2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
3. Click **Layouts** and select either a blank page or a predefined page layout, such as a page with a header and footer.

   Note: Predefined page layouts use panels to create columns, headers, and footers. These panels use inline CSS to set their position, so you can easily modify the layout after the page is created. However, if you're familiar with CSS and prefer using CSS rules, you can remove the inline style by selecting the panel, deleting the code from the Code tab in the Style pane ( ), and clicking **Apply**.
4. Choose a layout mode:
   - To expand the page to fill the width of the browser window, click **Full width**.
   - To set the page width, click **Fixed width** and enter the width.
5. Click **Create**. The page template opens.

Next, you must complete the template.

Tip:

- By default, any template you create is only available to other site administrators and designers in your organization. To let contributors create pages based on the template, select **Available to Contributor** in the Properties pane ( ).
- You can also create templates by converting or duplicating other pages.

Creating a Child Template

To use an existing template as a base for a child template:

- The quickest option is to:
  1. Select the template in the Page Templates view on the Overview tab and click > **Create Child Template**. Alternatively, click **Page Actions** > **Create Child Template** if the template is open.
  2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
  3. Click **Create**. The child template opens.
- Alternatively:

1. Hover over **Page Templates** on the Overview tab and click **New**, or click **New Page Template** in the Page Templates view.
2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
3. Click **Page templates** and select the page template.
4. Click **Create**. The child page template opens.

Completing Your Template

Once you've created a template, you must take the next steps to complete it:

- Lay out the page template
- Add other page elements to the template
- Create editable areas
- Create template-based site pages

SEE ALSO:

Creating Editable Template Areas
Identifying Which Template a Site.com Page Uses
Site.com Page Templates Overview
Setting Up the Contributor's Studio View

Creating Editable Template Areas

As the template creator, you specify which elements users can edit in pages based on the template.

To make a page element editable in derived pages or child templates, select it on the page template or in the Page Structure pane ( ) and click > **Make Editable**, or select **Editable** in the Properties pane ( ).

When the page template is open, editable page elements are highlighted with a blue border on the page. They also display a pencil icon ( ) in the Page Structure pane and in the information popup that appears when you hover over the element on the page.

SEE ALSO:

About Editable Page Elements
Setting Up the Contributor's Studio View
Creating Site.com Page Templates

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

About Editable Page Elements

When you mark a page element as `Editable` on a page template, that page element becomes editable in any child pages or templates derived from the parent template.

Consider these tips when creating editable page elements.

- Editable page elements are highlighted with a blue border in child pages and templates.
- If you make a page element within a panel editable, you can't also make the container panel editable.
- Users can't alter the events, style, or properties of an editable page element in pages based on the template.
- Users can't resize, reposition, or delete editable page elements in pages based on the template. However, if the element's Auto Height property is enabled in the template, its height will adjust to fit the content of the template-based page.
- Deleting an editable element from a page template removes it from all child pages and templates.
- When you enable the `Editable` property of a page element, any pages or child templates based on the template also inherit the enabled status. In turn, the enabled status in the child template cascades to any of *its* children, and so on. If you don't want its editability to cascade to any lower levels, disable the `Editable` property of a page element in a child template.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Editable Page Elements for Contributors

- Contributors can modify editable content blocks in site pages based on the template. They can also edit content blocks that you place in an editable panel in template-based site pages.

  Tip: To add a content block that only other site administrators or designers can edit, use custom code instead.

- Contributors can add content blocks to editable panels in site pages based on the template. If you make widgets available to contributors, they can also add them to editable panels.
- Site administrators and designers can edit any page element you make editable.

Default Content in Editable Page Elements

The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template won't trickle down to its children. (To return control of the content to the parent template, select the editable page element on the page or in the Page Structure pane and click > **Revert to Parent Content**. When you do this, any custom content in the editable page element is lost.)

Disabling the `Editable` property of a panel in a parent template overrides any changes made to that panel in child pages or templates. Changes to the panel at the child level disappear, and the panel reflects only the content from the parent template. However, the changes

at the child level aren't lost. Re-enabling the `Editable` property of the panel in the parent template restores the custom content previously added to its children. Any changes made to the element at the parent level will no longer show up.

SEE ALSO:

- Creating Editable Template Areas
- Setting Up the Contributor's Studio View
- Creating Site.com Page Templates
- Editing and Working With Site.com Page Elements

Creating Site.com Pages

As a site administrator or designer, when you create a site page, you can choose to base it on a page template. If you're creating several site pages that have common page elements, such as a navigation menu, you can save time and effort and achieve a consistent look and feel by creating a page template first and then basing your site pages on it. Alternatively, if none of your site pages have a similar structure or if you need to create a one-off site page that doesn't follow the overall site design, such as a home page, you can create a page based on a basic layout.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating Site Pages from a Layout

Start from scratch with a completely blank page or use a basic page layout.

1. Hover over **Site Pages** on the Overview tab and click **New**, or click **New** > **Site Page** when the Site Pages view is open.
2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Layouts** and select either a blank page or a predefined page layout, such as a page with a header and footer.

   Note: Predefined page layouts use panels to create columns, headers, and footers. These panels use inline CSS to set their position, so you can easily modify the layout after the page is created. However, if you're familiar with CSS and prefer using CSS rules, you can remove the inline style by selecting the panel, deleting the code from the Code tab in the Style pane ( ), and clicking **Apply**.
4. Choose a layout mode:
   - To expand the page to fill the width of the browser window, click **Full width**.
   - To set the page width, click **Fixed width** and enter the width.
5. Click **Create**. The site page opens.

Creating Site Pages from a Page Template

If you created a page template, you can base your site pages on it.

The quickest option is to:

1. Select the template in the Page Templates view and click > **Create Page from Template**. Alternatively, click **Page Actions** > **Create Page from Template** if the template is open.

2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Create**. The site page opens.

Alternatively:

1. Hover over **Site Pages** on the Overview tab and click **New**, or click **New** > **Site Page** when the Site Pages view is open.
2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Page templates** and select the page template.
4. Click **Create**. The site page opens.

Tip: You can also create pages by converting or duplicating other pages.

SEE ALSO:

Editing Site.com Pages as a Designer or Site Administrator

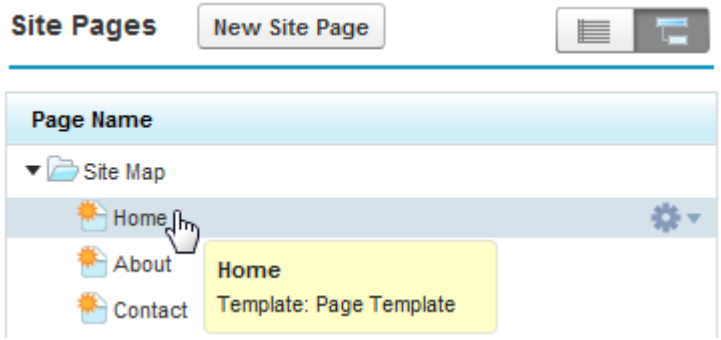
Adding Site.com Page Elements

Identifying Which Template a Site.com Page Uses

When you edit a template-based page, you can't modify its non-editable page elements. You also can't reposition, resize, or delete editable page elements, or alter the events, properties, or style associated with them. To update these elements or properties, you must edit them in the template the page is based on.

To identify which page template a site page is based on:

- Hover over the site page in the Sites Pages view on the Overview tab. An information popup appears that displays the page template's name.



- Examine the Page Structure pane when the page is open. The template's name is displayed as a link that you can click to open the template.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

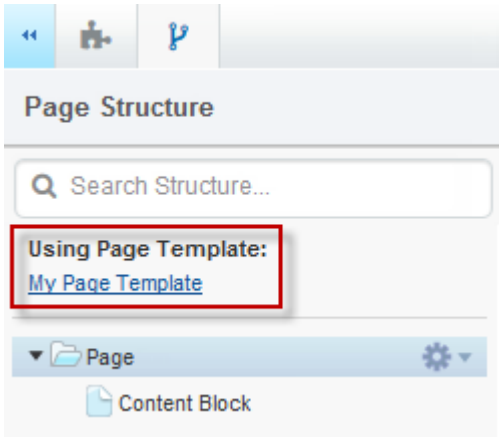
USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level



Tip: To view and open the site pages associated with a particular page template, select or hover over the page template in the Page Templates view of the Overview tab and click > **Edit Pages Based on Template**. Click a listed site page to open it.

SEE ALSO:

Creating Site.com Page Templates

Site.com Page Templates Overview

Renaming, Duplicating, and Converting Pages

When working with pages and templates, you can perform common tasks, such as renaming, deleting, or duplicating pages.

To access more page options, select or hover over a page on the Site Pages view of the Overview tab and click .

To access more template options, select or hover over a template in the Page Templates view of the Overview tab and click .

Alternatively, if the page or template is open, click on the toolbar.

The available options vary for pages and templates:

Select	To...
Edit	Open the page or template for editing. Alternatively, double-click the page or template.
Rename	Change the page or template name.
Preview	View the page in a browser window.
Duplicate	Create a copy of the page or template. Duplicating a page template doesn't duplicate the pages or templates based on it.
Delete	Remove a page or template. You can't delete a template that has pages based on it.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Select	To...
Create Child Template	Create a child template based on the selected template.
Create Page from Template	Create a page based on the template.
Convert Site Page to Template	Change the page into a template.
Convert Template to Site Page	Change the template into a page. You can't convert a template that has pages based on it.
Edit Pages Based on Template	View and open the site pages that are based on the selected page template.
Add IP Restrictions	Control access to the page or template by restricting the range of permitted IP addresses. When you create a page that's based on a template with IP restrictions, the page inherits the IP restrictions.

SEE ALSO:

- Editing Site.com Pages as a Designer or Site Administrator
- Using Site.com Studio as a Site Administrator or Designer

Changing a Page's Doctype Property

The Document Type Definition (DTD) or *doctype* of a page defines which version of HTML it's using. This information is used by some browsers to trigger a standard rendering mode. By default, each page's doctype is set to HTML5, which is the latest version, but you can change it to XHTML 1.0.

When the page is open:

1. Select the page in the Page Structure pane.
2. In the Properties pane, select an option in the Doctype drop-down list.

SEE ALSO:

- Changing a Page Element's HTML Tag
- Adding Custom HTML Attributes
- HTML5 Semantic Page-Layout Tags

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Features and Page Elements

Site.com Page Elements

Page elements are the building blocks of your site pages and page templates. Combined, they provide the page's structure and content.

When a site page or page template is open in Site.com Studio, these page elements are available to site administrators and designers.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Page Element	Description
Content Block	Contains the page text, and can also house images, media, and hyperlinks. Also available to contributors if the page has editable areas.
Custom Code	Lets you customize your site by adding markup, such as HTML and JavaScript for elements that aren't provided in Site.com Studio.
Image	Adds images directly to the page.
Breadcrumb	Adds a breadcrumb navigation element to the page.
Menu	Creates a menu that lets users navigate through the pages of your site.
Panel	Adds structure to the page and lets you group other page elements together.
Button	Adds a button to the page. You can use the actions in the Events pane to add functionality to the button.
Form	Lets you create web-to-lead forms or gather customer feedback, and submit the data to Salesforce objects.
Input Fields	Provides several field types to add to forms or pages. When added to a form, binds to fields in the form's object. See Input Field Types on page 773.
Data Element	Must be contained in a data repeater. Binds to a field in the data repeater's object and acts as a placeholder that shows the content of a specified field for the current record.
Data Function	Connects to a standard or custom Salesforce object, performs calculations on the returned results, and displays the calculation on the page.
Data Repeater	Connects to a Salesforce object and returns a dataset based on filters that you specify.

Page Element	Description
	Combines with data elements or custom code page elements to display the results on the page.
Data Table	Connects to a standard or custom Salesforce object, retrieves a data set based on the filter criteria that you specify, and displays one or more record as rows in the table.

SEE ALSO:

- Adding Site.com Page Elements
- Editing and Working With Site.com Page Elements

Adding Site.com Page Elements

Page elements are the building blocks of your site pages and page templates. Panel elements add structure to your pages. You can think of both pages and panels as "containers" for the page elements that you add to them.

If a site page or page template is based on another page template, you can only add page elements to editable panels, which are highlighted with a blue border on the page.

Pages, panels, data repeaters, and forms are "container" page elements, so you can add other page elements to them when the page is open.

- In the Page Elements pane ( ), either:
  - Drag the page element onto the page canvas or container page element.
  - Click the page element, select where to place it in the popup that appears, and click **Apply**.
- In the Page Structure pane ( ), hover over a container page element and click > **Add Page Elements**. Click the item you want to add or drag it onto the container page element.
- Select a container page element on the page and click > **Add Page Elements**. Click the item you want to add or drag it into the container page element.

When you drag a page element into an editable panel, the page element displays a permitted icon and a green border shows where you're placing the element.

If you try dragging a page element into a panel that isn't editable, the page element displays a not-permitted icon.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

Editing and Working With Site.com Page Elements
Adding Images Directly to the Page
Adding Content Blocks to Pages
Adding Custom Code to Pages
Adding a Navigation Menu

Editing and Working With Site.com Page Elements

The Page Structure pane ( ) displays the hierarchy of all elements on the page and is a very useful way of selecting, moving, and reordering elements, particularly for more complicated page designs.

- To select a page element, either click it on the page, or select it in the Page Structure pane. This highlights the element on the page and displays the item's selector bar and **Actions** menu ( ).
- To edit a page element, such as a content block, image, or custom code, either:
  - Double-click the element on the page.
  - Select the element on the page and click > **Edit** on its selector bar.
  - Select or hover over the element in the Page Structure pane and click > **Edit**.
  - Click **Edit Content** on the toolbar (for content blocks only).
- To move a page element, either:
  - Select the element on the page and drag it to the correct position, or click > **Move** on its selector bar.
  - Drag the item to your preferred location in the Page Structure pane, or select it and click > **Move** > *Direction*. You can also drag all page elements other than panels to your preferred location in the Page Structure pane.
- To resize a page element, select it and drag the resize handles to the correct size. If the corner resize handles are grayed out, it means the item's Auto Height property is enabled, which adjusts the height depending on its contents. To resize it to a set height, disable the property by either deselecting **Auto Height** in the Properties pane ( ), or by clicking one of the bottom resize handles and clicking **Disable Auto Height** in the popup message that appears.

  Tip: If you disable the Auto Height property on an image, but you want it to retain its aspect ratio—the relationship of height to width—press and hold down the SHIFT key while you drag to resize it.

  Alternatively, if you're using CSS to style the page element, adjust the style of the class or ID that styles it.
- To delete an element, select it and either:
  - Click > **Delete** on the item's selector bar.
  - Press **DELETE**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

- Click  on the toolbar.
- In the Page Structure pane, click  > **Delete**.

If a site page or page template is based on another page template:

- The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template won't trickle down to its children.
- You can't reposition or resize its page elements. However, if the element's Auto Height property is enabled in the template, the height will adjust to fit the content in the template-based page. To edit the page element, you must edit it in the page template.
- You can't delete its page elements. To delete the page element, you must delete it from the page template.
- You can't alter the events, properties, or style of an editable page element, such as its color, position, and size.

SEE ALSO:

Site.com Page Elements

Adding Site.com Page Elements

Adding Content Blocks to Pages

Laying Out Site.com Pages Using Panels

About Editable Page Elements

Working with Panels

Laying Out Site.com Pages Using Panels

A panel is a useful layout tool that defines the logical divisions of your page and lets you group page elements together for easy movement and positioning. Think of it as a container for other page elements, including other panels, or as a `div` that wraps around the content placed within it. Panels are ideal for adding editable areas to page templates.

When you create a page template or site page, you can use predefined page layouts that include headers, footers, and columns, which are created using panels. Once created, you can then further modify the layout to match your site's design.

If you need to add more divisions to the page and you're not familiar with CSS, the easiest method is to use row and column panels. This feature adds panels with predefined CSS positioning to ensure they align correctly on the page.

Note: Predefined page layouts, and row and column panels use inline CSS to set their position. If you're familiar with CSS and are using CSS rules to style your site, you can remove the inline CSS by deleting it from the Code tab in the Style pane ( ) and clicking **Apply**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To add row and column panels to a page:

1. Select the page (the top folder icon) in the Page Structure pane ( ).
2. Click > **Add Rows and Column Panels**.
3. Select the number of row or column panels you require. If the page already contains content, it is placed in the first new panel.

To add a row panel:

- Above a panel, select the panel on the page or in the Page Structure pane and click > **Add Rows and Column Panels** > **Insert Row Above**
- Below a panel, select the panel on the page or in the Page Structure pane and click > **Add Rows and Column Panels** > **Insert Row Below**

To add row and column panels to another panel:

1. Select the panel on the page or in the Page Structure pane.
2. Click > **Add Rows and Column Panels**.
3. Select the number of row or column panels you require. If the page already contains content, it is placed in the first new panel.

To add a single panel, drag a **Panel** from the Page Elements pane ( ) onto the page.

By default, the height of a panel automatically adjusts when you add content to it because its Auto Height property is enabled. You can disable the property to resize and reposition panels. If you hover over a panel on the page, an information popup appears that displays the width and height of the panel.

When you drag a page element onto a panel, the edges change color, indicating that the element is now grouped within it. To remove the element from the group, drag it outside the panel.

SEE ALSO:

- Adding Site.com Page Elements
- Positioning Panels Using CSS
- Changing a Page Element's HTML Tag

Positioning Panels Using CSS

A panel is a useful layout tool that defines the logical divisions of your page. Using CSS, you can position panels and improve the layout of the page.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Adding Padding and Margins to Panels

Two CSS properties—margins and padding—can help with your page layout by creating space between the rows and columns, and the content within. The margin property controls the space outside the panel between its border and outer edge, while the padding property controls the space between the panel's content and border.

To add margins and padding:

1. Select the panel.
2. Open the Dimensions section of the Style pane.
3. In the Margins section, either:
   - Set the margin width for all four sides by entering a value in the `All` text box and selecting the unit of measurement.
   - Set the margin widths for the top, right, bottom, or left sides independently by entering a value in the appropriate text box and selecting the unit of measurement.
4. Similarly, in the Padding section, set the padding widths as required. Adding padding increases the total width of the panel. For example, if you have a panel with a width of 500px and you add padding of 20px to all sides, the total width of the panel will be 540px.

Tip: You can center a panel or block page element using the margin property. Enter *0* in the `All` text box and select Auto in the drop-down list.

Creating Column Panels Using the Float Property

If you need to add more divisions to the page and you're not familiar with CSS, the easiest method is to use row and column panels. Alternatively, using the CSS float property, you can position panels to the left or right to create columns. (When you add panels using the row and column panels feature, they're automatically positioned using the float property.) For example, you could add two single panels to a container panel and set both panel's float and width properties to create a two-column page layout.

To create a column panel:

1. Select the panel.
2. Open the Layout section of the Style pane.
3. Click  to float the panel to the left, or click  to float the panel to the right. If you're creating a two-column layout, for example, ensure you set the float property of both panels.

4. Adjust the width of the panel to ensure the panels align correctly by either setting the width in the Dimensions section or dragging the panel's border on the page. For example, if you're creating two columns of equal width, set the width of both panels to 50%.

Tip: When you use the float property, remember to set the overflow property of the container panel to "hidden." This allows the container panel to grow as the height of the column panels increase. Select the container panel and in the Layout section of the Style pane, select Hidden in the Overflow drop-down list.

SEE ALSO:

Cascading Style Sheets Overview

Laying Out Site.com Pages Using Panels

Adding Images Directly to the Page

As a site administrator or designer, you can add images directly to your site pages and page templates or you can add images to content blocks.

To add an image directly to the page:

1. Open the site page or page template.
2. Drag an **Image** from the Page Elements pane onto the page.
3. In the Add an Image dialog box, either:
   - Find an existing image by typing its name in the `Search Image` text box and selecting it from the list.
   - Upload an image from your computer in the Upload tab by browsing to the image, clicking **Upload**, and selecting it from the list.
4. Click **Apply**. The image is added to the page.
5. Enter a brief description of the image in the `Alternative Text` field in the Properties pane. The description is used by screen reader users or as a substitute if the browser can't display the image. It can also help with search engine optimization (SEO).

SEE ALSO:

Adding Site.com Page Elements

Editing and Working With Site.com Page Elements

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Working with Content Blocks

Adding Content Blocks to Pages

Content blocks contain the text of your website pages, and can also house images, videos, and hyperlinks. Designers and site administrators can add content blocks to pages when in Design Mode.

To add a content block when the page is open, either drag it from the Page Elements pane onto the page or target a container page element.

To edit a content block, double-click it. For greater control over the text, you can edit the HTML directly by selecting the content block and clicking > **Edit HTML**.

If you make a content block in a page template editable, contributors can edit the content block in any page based on the template. To add a content block that only other site administrators or designers can edit, use custom code instead.

SEE ALSO:

- Adding Images to Content Blocks in Design Mode
- Attaching Hyperlinks to Text and Images in Design Mode

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Understanding the Content Editing Toolbar

Designers and site administrators can edit content blocks when in Design mode. Content blocks contain the site page's text, along with images, videos, and hyperlinks.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When in Design Mode, you can use the content editing toolbar to:

- Undo and redo your edits, and remove the formatting of text copied and pasted from Microsoft® Office, which can often include hidden formatting (1).
- Cut, copy, and paste text (2).
- Apply direct formatting (3), such as:
  - Font family and size
  - Bold, italic, underline, subscript, superscript, and strikethrough
  - Font and highlight color
- Control the text style and layout (4) by:
  - Applying paragraph and heading styles
  - Setting paragraph indentation
  - Left-aligning, centering, right-aligning, or justifying text
  - Inserting numbered or bulleted lists
- Insert a table, and add rows, columns, and spacing (5).
- Add images, videos, and special characters (6).
- Add and remove hyperlinks and anchors (7).

Tip: Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. This also ensures that all page text is updated automatically if a site administrator or designer modifies the site's paragraph and heading styles.

Editing Content Blocks in Design Mode

Designers and site administrators can edit content blocks on the page. Content blocks contain the text for site pages, and can also house images, videos, and hyperlinks.

When the page is open in Design Mode:

1. Double-click the content block on the page.
2. Add or edit text and format it using the content editing toolbar.

   Tip:

   - If you copy and paste text from Microsoft® Office, highlight the text and click  to remove any hidden formatting, which can adversely affect how the text appears on the page.
   - Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. This also ensures that all page text is updated automatically if a site administrator or designer modifies the site's paragraph and heading styles.

3. Add images, videos, hyperlinks, or anchors as required.
4. Click **Save**.

Note: The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template won't trickle down to its children.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Images to Content Blocks in Design Mode

Designers and site administrators can add images to content blocks when viewing a page in Design Mode.

When the page is open:

1. Double-click the content block on the page.
2. Position your cursor where you want to insert the image and click  .
3. In the Image Properties dialog box, either:
   - Enter a URL to an image in the `Image URL` field.
   - Select an image from your website by clicking **From Website** and selecting the image in the list that appears.
   - Upload an image from your computer by opening the Upload tab, browsing to the image, and clicking **Upload**.
4. Enter a brief description of the image in the `Alternative text` field. The description is used by screen reader users or as a substitute if the browser can't display the image. It can also help with search engine optimization (SEO).
5. Optionally, preview how the image appears in relation to the text on the page and set:
   - The width and height of the image
   - How much space surrounds the image (which is controlled by the `HSpace` and `VSpace` properties)
   - How it aligns with the text on the page
   - The image border (for example, to set a dotted green border that's 10 pixels wide, you enter `10px dotted green` in the `Border` field)
6. Click **Apply**.
7. Click **Save**.

SEE ALSO:

Editing Content Blocks in Design Mode

Understanding the Content Editing Toolbar

Importing and Managing Assets

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Adding Video to Content Blocks in Design Mode

Designers and site administrators can add YouTube®, Google®, Adobe® Flash®, Windows Media®, and Apple QuickTime® videos to content blocks when viewing a page in Design Mode.

When the page is open:

1. Double-click the content block on the page.
2. Position your cursor where you want to insert the video and click .
3. In the Video Properties dialog box, select the video type and either:
   - Enter the URL to the video in the `Video URL` text box—for example, `http://www.youtube.com/watch?v=123abc`.
   - Select a video from your website by clicking **From Website** and selecting the video in the list that appears.
   - Upload a video from your computer by opening the Upload tab, browsing to the image, and clicking **Upload**.

     Note: You can only select or upload Flash, Windows Media, and QuickTime videos.
4. To specify how the video is displayed on the page, you can set:
   - The width and height of the video
   - How much space surrounds the video (which is controlled by the `HSpace` and `VSpace` properties)
   - How it aligns with the text on the page

   You can also preview the video.
5. Click **Apply**. The video appears as an icon in the content block.
6. Click **Save**.

Note: You can view all video types, other than Windows Media videos, when you preview a page.

SEE ALSO:

Editing Content Blocks in Design Mode

Understanding the Content Editing Toolbar

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Attaching Hyperlinks to Text and Images in Design Mode

When viewing a page in Design Mode, designers and site administrators can create hyperlinks to:

- External Web pages or websites
- Pages and assets in the site
- Email messages
- Anchors on the page

When the page is open:

1. Double-click the content block on the page.
2. Select the text or image that you want to attach a hyperlink to and click  .
3. Select the link type in the Link to drop-down list. To link to:
   - A Web page:
     a. Select **A URL**.
     b. Type the address in the `URL` field—for example, `http://www.externalsite.com`.
     c. Go to step 4.
   - A page or item in your site:
     a. Select **An item in your site**.
     b. Select the item type, such as a page or image.
     c. Select the item. (If you can't see the list of items, place your cursor in the `URL` field and press the DOWN key on your keyboard.)
     d. Go to step 4.
   - An anchor that you previously added to the page:
     a. Select **An anchor on the page**.
     b. Select the anchor in the drop-down list. Alternatively, enter a new anchor name and create the anchor afterwards.
     c. Go to step 5.
   - An email message:
     a. Select **An email**.
     b. Enter the recipient's email address and the message information.
     c. Go to step 5.

   Note: For content blocks in a data repeater, you can use expressions to add a custom link, such as a URL query string, to an item in your site or to a Web page.
4. To select which window the item should open in, select an option in the Target drop-down list:
   - **Popup window** loads the item into a popup window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
   - **New window (_blank)** loads the item into a new, unnamed browser window.
   - **Same window (_self)** loads the item into the same frame or window as the link. This is the default setting.
   - **Topmost window (_top)** loads the item into the topmost parent frameset or window of the frame that contains the link.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

- **Parent window (_parent)** loads the item into the parent frameset or window of the frame that contains the link.

5. Optionally, enter a tooltip description for the link. The tooltip displays as a pop-up when the user hovers over the link.
6. Click **Apply**.
7. Click **Save**.

To delete a hyperlink, select it and click .

SEE ALSO:

Editing Content Blocks in Design Mode
Understanding the Content Editing Toolbar

Adding Anchors to Pages in Design Mode

An anchor is an invisible marker that identifies a particular location on a page. If you're a designer or site administrator, you can add an anchor to the page and then create a hyperlink that jumps to that specific location. This can be useful if a page is particularly long; for example, if your page has several sections, you could add links to each section at the top of the page to aid navigation.

When the page is open in Design Mode:

1. Double-click the content block on the page.
2. Place your cursor at the beginning of the line where you want to link to and click .
3. Enter a name for the anchor and click **Apply**. Ideally, use a name that helps identify the anchor's location on the page—for example, *top*.
4. Now create a hyperlink that links to the anchor.

SEE ALSO:

Attaching Hyperlinks to Text and Images in Design Mode
Editing Content Blocks in Design Mode
Understanding the Content Editing Toolbar

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Adding Custom Code to Pages

Custom code lets you customize your site using markup, such as HTML and JavaScript.

- Add markup to a specific location on a page using the Custom Code page element. JavaScript added using the Custom Code page element loads when that part of the page loads.
- Add markup to the page `head`. JavaScript in the page `head` loads first.
- Add JavaScript to the page `body` on page 714. JavaScript added to the page `body` is positioned at the end of the `body` tag and only loads when the DOM is ready.
- Add a reference to a JavaScript file or library on page 714 in the page `head` or `body`.

Tip:

- Scripts can't execute while you're editing a page in Site.com Studio. To test your code, preview the page.
- If you are building a Site.com site from an existing HTML site, avoid using the Custom Code page element to paste large chunks HTML from the original site. Instead, use the available page elements, such as panels, content blocks, and data tables. This will let you make future updates and design changes much more easily.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Adding Markup Directly to the Page

1. Drag a **Custom Code** page element from the Page Element pane onto the page.
2. Enter the code in the Edit Code dialog box.
3. Click **Save and Close** to add the code directly to the page.

Adding Markup to the Page `Head`

1. In the Scripts section of the Properties pane, click **Configure** in the Edit Head Markup section.
2. Enter the markup in the Edit HTML Code dialog box.
3. Click **Save and Close** to insert the markup into the page `head`.

Adding JavaScript to the Page `Body`

1. In the Scripts section of the Properties pane, click **Configure** in the Edit Body Scripts section.
2. Enter the code in the Edit JavaScript Code dialog box. Don't add `<script>` tags, as they're already included.
3. Click **Save and Close** to add the code to the bottom of the page `body`.

Using JavaScript Files or Libraries

Instead of adding JavaScript code directly to a page, you can include links to imported or external JavaScript files, or to an open-source library (via the Google Libraries API).

1. In the Scripts section of the Properties pane, click + in either the Body Scripts or the Head Scripts section.
2. To link to:
   - A JavaScript file that you've imported, select **An imported script** and select the file.
   - An open-source JavaScript library, select **A Google AJAX library** and select the library.
   - An external JavaScript file, select **A URL to an external script** and enter the address.

3. Click **Apply**.

SEE ALSO:

Displaying Data Using Custom Code

Adding Site.com Page Elements

Adding Website Navigation

About the Site Map and Page Hierarchy

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

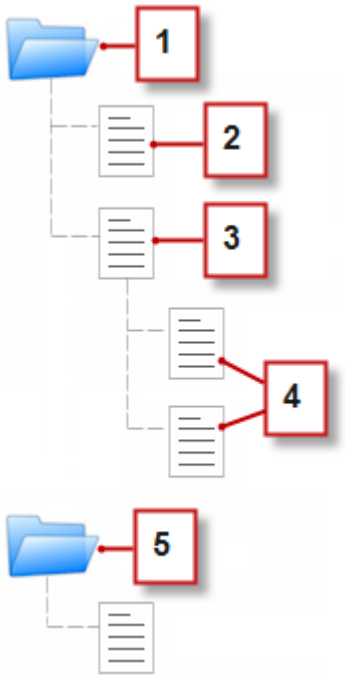
Available (with limitations) in: **Developer** Edition

The Site Pages view on the Overview tab contains the pages and site map links of your website. The Site Map folder reflects the hierarchy or tree structure of your site by housing site pages and links that are included in the site map. When you create new pages or site map links, they're automatically added to this folder. The Landing Pages folder houses standalone pages that are excluded from the site map, making it ideal for temporary pages, such as promotional or competition pages.

Tip: If you can't see the Site Map folder in the Site Pages view on the Overview tab, click .

When adding a navigation menu to your site, it's important to organize the hierarchy of your site pages and links accurately, because this structure is used to generate the menu. Pages and site map links are displayed in navigation menus in the order you arrange them.

In this representation of the site hierarchy, you can more clearly see the tree structure.



1. The Site Map folder, which contains four site pages.
2. A top-level page in the site hierarchy.
3. A top-level, parent page with two child pages. A child page is a page at a lower level in the site hierarchy than its parent page.

4. Two child pages.
5. The Landing Pages folder, which contains a temporary page that's not part of the site map or navigation menu.

Tip: If a page has child pages, the ▶ icon appears beside it indicating that you can expand the branch.

By default, when you create a menu, it's generated from the pages and site map links in the Site Map folder in the Site Pages view. However, you can also create a menu that's generated from the pages in the Landing Pages folder or from the child or sibling pages of a site page.

You can hide a page in menus, breadcrumbs, and the site map by selecting the `Hide Page` checkbox found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

SEE ALSO:

- Adding a Navigation Menu
- Creating Site.com Pages

Adding Links to a Site Map

Pages aren't the only things you can assign to your site map. Add internal or external URLs to your site map to customize your navigation menus and breadcrumbs, and achieve greater flexibility and control.

For example, let's say you have a top-level Products page that has a menu based on its child pages. You also want to include a page called Testimonials in the menu, but it's not a child page of the Products page. You can create a site map link that points to the Testimonials page and add the link under Products in the site map. Now when visitors view the Products page, they see a menu consisting of its child pages, along with a menu item that takes them directly to Testimonials.

1. On the Overview tab, click **New** > **Site Map Link**.
2. Enter a name for the link.
3. Enter a URL. URLs can be either relative or absolute, and are case sensitive.

   Note: You can't preview absolute site map links in Site.com Studio unless they include a prefix, such as `http://` or `https://`.

4. Click **Create**.
   The link appears at the bottom of the site map.
5. Drag the link to the correct position in the site map.

Note: Site map links are automatically included in navigation menus and breadcrumbs. However, you can't set a site map link as a custom root node in a breadcrumb.

SEE ALSO:

- Adding Breadcrumb Navigation to Pages
- Adding a Navigation Menu
- About the Site Map and Page Hierarchy

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Breadcrumb Navigation to Pages

Add a Breadcrumb page element to your page to help users navigate through your site and show the page's location in the site hierarchy.

Breadcrumb navigation is based on the pages and links in the site map and usually supplements menu navigation. You can hide a page in menus, breadcrumbs, and the site map by selecting the `Hide Page` checkbox found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

Tip: To save time and effort, add a Breadcrumb page element to a page template to automatically include it on every template-based site page. The breadcrumb on each derived page dynamically updates based on its location in your site map.

1. Open the page or page template that you want to add the breadcrumb element to.
2. Drag a **Breadcrumb** from the Page Elements pane onto the page.
3. In the Properties pane, under Root, specify a custom root node for the breadcrumb. By default, the root value is set to None, which builds the breadcrumb structure based on the page's location in the site map.
   - Select *`Home Page`* to set the site's home page as the first item in the breadcrumb. To find out how to set a site's home page, see Configuring Site Properties.
   - Select a specific site page to set it as the first item in the breadcrumb.

   Note: You can't set a site map link as a custom root node in a breadcrumb.
4. In the Properties pane, under `Separator`, you can customize the separator used between breadcrumb nodes.

   By default, the separator is >. However, you can change it to another text symbol or insert HTML code for an image in your site, such as `<img src='/separatorimage.jpg'/>`.
5. To style the breadcrumb:
   a. Click the Style pane and ensure `Class` is selected.
   b. Choose an option from the Style drop-down list.
   c. Adjust the values in the Visual tab as desired.

SEE ALSO:

Adding a Navigation Menu

Editing and Working With Site.com Page Elements

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding a Navigation Menu

By default, when you create a menu, it's generated from the pages and site map links in the Site Map folder in the Site Pages view. However, you can also create a menu that's generated from the pages in the Landing Pages folder or from the child or sibling pages of a site page.

1. Arrange site pages and site map links in the Site Map folder or the Landing Pages folder in the order you want them to appear in the menu by dragging them to the desired location. Drag a page or site map link onto another page or link to make it a child of that item. Drag pages or links into the Landing Pages folder to exclude them from the site map.

   Tip: If you can't see the Site Map folder in the Site Pages view on the Overview tab, click .

2. Open the page template or site page that you want to add the navigation menu to.
3. Drag a **Menu** from the Page Elements pane onto the page.
4. In the Properties pane, under Menu Source, select the pages that you want to use for the menu. By default, the Site Map folder is used to create the menu. Any site map link in the applicable hierarchy also shows up in your menu.
   - Select *`Landing Pages`* to create the menu from the pages in the Landing Pages folder.
   - Select *`Child Pages`* to create the menu from the current page's child pages.
   - Select *`Sibling Pages`* to create the menu from all of the pages that share the same parent as the current page.
   - Select a specific site page to create the menu from just its child pages.

   Note: If you add a menu element to a page template, the menu doesn't display correctly in the template if you select *`Child Pages`* or *`Sibling Pages`* as the menu source, because page templates aren't part of the site map hierarchy. However, the menu appears as expected on site pages based on the page template.

5. To alter the appearance of the menu, you can select a different theme from the Theme Name drop-down list. For example, to create a drop-down menu, select `Horizontal Drop-down`. You can modify the style of any theme to suit your needs.
6. To change the name of a page in the menu, open the associated page and update its `Navigation Name` field in the Properties pane. Navigation names can include spaces and special characters.

   Alternatively, to change the name of a site map link in your menu, hover over the link in the Site Pages view on the Overview tab, click **Edit**, and update the name.

Tip:

- When you add a new page or site map link, update a page's `Navigation Name` property, or rearrange pages or links, the menu updates automatically to reflect the changes.
- To automatically include a menu on every site page, add the menu to a page template and base the site pages on it.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

- You can hide a page in menus, breadcrumbs, and the site map by selecting the `Hide Page` checkbox found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

SEE ALSO:

Styling Navigation Menus

About the Site Map and Page Hierarchy

Adding Site.com Page Elements

Styling Navigation Menus

Navigation menus are styled using CSS themes that you can customize to match the design of your website. When you add a navigation menu to a page, it uses a default theme to control its appearance. Choose from other existing themes in the Theme Name drop-down list in the Properties pane.

Alternatively, to customize a theme to suit your needs:

1. Select the navigation menu on the page.
2. Select a theme to use as a base in the Theme Name drop-down list in the Properties pane. Use a theme that most closely matches your site design or select *`Blank`* to start with a completely blank theme.
3. Open the Style pane and ensure `Class` is selected.
4. In the Style drop-down list, select the part of the menu that you want to style. When you select an item, it's highlighted for a few seconds, so you can easily see which part you're styling.

   Tip: If you're familiar with CSS, you can also modify the style of the menu in the site's style sheet.

5. To style the selected menu item, use the Style pane properties. Your changes are immediately reflected in the menu.
6. Repeat as required for each part of the menu.

SEE ALSO:

Adding a Navigation Menu

About the Site Map and Page Hierarchy

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Custom HTML Attributes

You can add custom HTML attributes to pages and page elements, which are rendered on the HTML tag of the page element. For example, this is useful when working with third-party frameworks that render page elements differently based on certain attributes.

When the page is open:

1. Select the relevant page or page element in the Page Structure pane.
2. In the HTML Attributes section of the Properties pane, click  .
3. Enter a name and value for the HTML attribute.
4. Click **Save**.

To delete an HTML attribute, select it and click  .

To change the order in which an HTML attribute is rendered, select it and click  or  .

SEE ALSO:

- Changing a Page Element's HTML Tag
- HTML5 Semantic Page-Layout Tags
- Changing a Page's Doctype Property

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Changing a Page Element's HTML Tag

By default, panels, data repeaters, data elements, custom code, and content blocks are each defined as a `div`, but you can change this to any other HTML tag using the `HTML Tag` property. This gives you greater flexibility and control over how the page element is displayed on the page.

Warning: The `HTML Tag` property provides a powerful way to control how page elements are displayed. However, if you change a page element's HTML tag, you may generate invalid HTML. Before publishing any changes, test the page thoroughly.

To redefine a panel, data repeater, data element, custom code, or content block:

1. Select the element on the page.
2. In the `HTML Tag` field in the Properties pane, start typing the tag name.
3. In the auto-complete list that appears, select the relevant HTML tag. Alternatively, you can define your own HTML tag—for example, if you're working with a JavaScript library or if new HTML5 tags are introduced in the future. You can also remove the HTML tag on a panel, data repeater, data element, custom code, or content block to disable its ID, class, or inline styles.

Note: The following tags aren't included in the auto-complete list:

- `base`
- `body`
- `doctype`
- `head`
- `html`
- `meta`
- `style`
- `title`

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit page element properties:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned in Site.com Studio

SEE ALSO:

HTML5 Semantic Page-Layout Tags

Adding Custom HTML Attributes

HTML5 Semantic Page-Layout Tags

HTML5 defines several semantic page-layout tags that describe the content they contain. These tags make it easier for search engines and screen readers to read and organize your content.

By default, several page elements are defined as a `div`, including panels, data repeaters, data elements, content blocks, and custom code. Using a page element's `HTML Tag` property, you can change the tag to a semantic HTML5 block tags, such as:

Option	Description
Article	A section containing an independent item of content, such as a magazine article or a forum post.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Option	Description
Aside	A section containing content that's only superficially related to the main page content, such as a sidebar or advertising.
Details	A section containing additional details that the user can view or hide using an interactive widget. It can also include a summary section.
Header	A section containing an introduction, or a group of navigation elements.
Footer	A footer section for the page or parent section. It typically contains information about the parent section and appears at the end of the section.
Nav	A section that contains navigation links.
Section	A generic section of the page.
Summary	A summary or caption section for a details section.

Tip: If you use a HTML5 semantic tag, it's good practice to also change the page's doctype to HTML5.

SEE ALSO:

Changing a Page Element's HTML Tag

Adding Custom HTML Attributes

Setting Up the Contributor's Studio View

Control what your contributors can do in Site.com Studio.

The contributor's Site.com Studio view lets users browse and update website content. To enable contributors to edit the site's content, however, you must first create template-based pages with editable areas and enable several properties.

To enable your contributors to:

- Edit the text on a page, create template-based pages with editable content blocks
- Add content blocks to the page, create template-based pages with editable panels
- Create site pages based on a page template, select **Available to Contributors** in the Properties pane when the page template is open
- Add widgets to the page, select **Available to Contributors** in the Widgets view on the Overview tab or in the Properties pane when the widget is open
- Update the appearance of pages and widgets, set up branding properties

To ensure that the contributor's Site.com Studio view is set up correctly, click **View Studio as a Contributor** in the site's pull-down menu (on the top toolbar). To exit, click **Return to My Studio View**.

The contributor's view is not available by default for Site.com Community sites. However, you can use a Site.com Contributor license to grant contributor access to a specific user. See *About Feature Licenses* in the Site.com help for details. Alternatively, a user can preview the Site.com Community site as a contributor by appending `?iscontrib` to the site's URL. For example: `https://sitestudio.na1.force.com/?iscontrib`

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

- Creating Site.com Page Templates
- Creating Widgets
- Site Branding Overview
- Understanding the Contributor's Page Editing View

Styling Your Website

Cascading Style Sheets Overview

Cascading Style Sheets (CSS) provide a flexible way to add style to the pages of your website. This collection of formatting rules governs the appearance of your pages, and lets you define the fonts, colors, layout, and other presentation features.

By using CSS to control your fonts, you can ensure greater consistency in the appearance and layout of your pages in multiple browsers. Some of the many text properties that CSS lets you control include font family and size, text and background color, text formatting, and link color.

Using CSS, you can also position, add color to, float text around, and set margins and borders for block-level elements. A block-level element is a standalone piece of content that's visually formatted as a block. For example, content blocks (which are equivalent to `p` tags) and panels (which are the same as `div` tags) are both block-level elements.

Site.com supports CSS3, which is the latest specification for CSS.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

About Inline Styles Versus Style Sheets

In Site.com Studio, you can:

- Apply styles directly to a selected page or page element using the **Inline** option in the Style pane ( ). Inline styles apply only to the selected item.
- Add style items such as CSS classes or IDs to a style sheet, and apply the style items to the selected page or page element. This approach separates the content (your web pages) from the presentation (the style sheet).

If you're not familiar with CSS, you'll probably find the inline option the easiest to use and understand. However, inline styles lose many of the advantages of style sheets because they mix content with presentation—the inline style is only applied to that individual element. If you need to update the style of your site, you have to update the style properties of every affected page and page element.

By contrast, although style sheets may be more difficult to understand at first, they enable you to make site-wide changes from one convenient location. When you update a style item in your style sheet, it immediately updates the style of every page or page element that uses it.

It's worth taking the time to become familiar with CSS because it:

- Saves you time and effort when building and designing your site
- Produces cleaner, more consistent site designs
- Simplifies navigation for people with accessibility issues (such as site visitors using screen readers)

For more information about using CSS and creating style sheets, go to the World Wide Web Consortium (W3C) at `www.w3.org/Style/CSS`. There are also many tutorials available on the Internet that provide in-depth CSS training.

About CSS Classes and IDs

When you use style sheets to style your site, you can redefine the formatting of HTML tags such as `body` or `h1`. You can also create CSS classes and IDs to define the style of particular elements, such as headers or repeating content. A CSS class lets you define and apply style properties to many elements on a page, whereas a CSS ID is ideal for targeting a single item on a page. For example, in a page's structure, IDs are often used to define the header and footer areas, as each page has only one header or footer, but classes are used to define repeating page elements, such as a blog post.

Best Practices

- Include a CSS reset in your style sheet to reset all style items to a baseline value. This helps avoid cross-browser differences due to their built-in default style settings.
- Use CSS classes and IDs instead of inline styles wherever possible. This promotes the separation of presentation and content, and makes it easier to updates the site's styles.
- Use IDs when there is only one occurrence per page. Once you've used the ID, you can't use it again on that page. Use classes when there are one or more occurrences per page.
- Use groups to organize your CSS logically and make it easier to maintain your style sheet.
- If you're using CSS3, ensure you preview and test your site in each browser that you want it to support, because some browsers haven't yet fully implemented CSS3 features.

SEE ALSO:

Using the Style Pane

Understanding the Style Sheet View in Site.com

Creating and Using CSS Style Sheets

Using the Style Pane

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements. If you're using CSS classes or IDs to style your pages, you can modify or create style items directly from the Style pane, rather than opening the style sheet.

To apply a style to a selected page or page item:

1. Open the Style pane ( ).
2. To apply:
   - An inline style, select **Inline**. Inline styles affect the selected item only and aren't included in a style sheet.
   - A CSS class, select **Class** and start typing the name. If the class already exists in your style sheet, select it in the list that appears. To create a new class, type the name, select it, and click **Yes** to add it to the style sheet.
   - A CSS ID, select **ID** and select it in the drop-down list. To create a new ID, click +, enter the ID name, and click .

   Menu page elements have several components, which you can style individually by selecting your preferences in the Style drop-down list that appears.
3. In the Visual tab, apply style properties as appropriate. Alternatively, in the Code tab, you can type your CSS styles directly and click **Apply**.

Tip: To view the style properties associated with a selected page or page element, open the Code tab of the Style pane. To remove the style properties, click **Clear**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

Creating Style Sheet Items and Groups

Creating and Using CSS Style Sheets

Cascading Style Sheets Overview

Style Pane Properties

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

The Background Section

Property	Description
Background Color	Sets the element's background color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Background Color drop-down list.
Background Image	Adds a background image to the element. Click **URL** and enter the image URL, or click + to select an imported image.

Property	Description
Background Repeat	Tiles the element's background image. •  ensures that only one copy of the image appears. •  repeats the image horizontally. •  repeats the image vertically. •  tiles the image both horizontally and vertically.
Position	Specifies the position of the element's background image. • To set the horizontal position of the background image, enter a value in the `X` text box and select a unit of measurement. Alternatively, select Left or Right in the drop-down list. • To set the vertical position of the background image, enter a value in the `Y` text box and select a unit of measurement. Alternatively, select Top or Bottom in the drop-down list.
Cursor	Sets the cursor type, such as crosshair or pointer.

The Font & Color Section

Property	Description
Font	Sets the font family of the selected element.
Color	Sets the font color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Color drop-down list.
Size	Sets the font size. Enter a value in the `Size` text box and select a unit of measurement such as em, point, or %. Alternatively, select a predefined value such as XX-Small. Select Inherit to use the same font size as the parent element (for example, the page or panel).  Tip: Use relative sizes such as em or a percentage to enable your end users to resize the font size in their Web browsers.
Style	Formats the element's font style. •  makes the font bold. •  makes the font bold and italic. •  makes the font italic. • **None** removes existing styles.
Font Variant	Specifies whether to render the font as small capitals.

Property	Description
Line Height	Modifies the amount of space between lines of text. Enter a value in the text box and select a unit of measurement such as pixels, percentage, or em. Select Inherit to use the same line height as the parent page element.
Text Decoration	Applies decorative effects to the element's text. For example, you could remove the underline that usually appears under hyperlinks, which is a standard CSS rule that's built in to most Web browsers. •  underlines the text. •  applies strikethrough formatting. •  displays a line over the text. • **None** removes existing text decoration.
Align	Aligns the text of the selected element. •  aligns the text to the left. •  aligns the text to the right. •  centers the text. •  aligns the text with both the left and right margins.
Case	Changes the capitalization of the element's text. •  capitalizes the first character of each word. •  capitalizes all characters. •  lowercases all characters. • **None** removes existing capitalization formatting.
Text Indent	Indents the first line of text of the selected page element. Enter a value in the text box and select a unit of measurement. Select Inherit to use the same indentation as the parent page element.
White Space	Controls how white spaces such as spaces, tabs, and hard returns are handled inside an element.

The Layout Section

Property	Description
Positioning	Positions page elements outside the normal flow of the document. Usually, elements on a page are rendered in Web browsers in the order they appear in the document. Block elements such as `p` tags and `div` tags appear one beneath the other, whereas inline elements such as `em`, `strong`, and `span` tags are rendered next to text or each other.

Property	Description
	• Absolute positions the content using the settings in the `Top`, `Bottom`, `Left`, and `Right` text boxes. • Relative renders the page element in the normal layout flow, but moves the element relative to its normal position depending on the values in the `Top`, `Bottom`, `Left`, and `Right` text boxes. For example, if you set an element's left position to 20 pixels, the page element will be positioned 20 pixels further to the left.
Display	Overrides a page element's default layout behavior. For example, you can hide page elements, make block elements render inline, or make inline elements render as blocks. • None hides the page element. • Block displays the page element as a block-level page element, with a line break before and after the element. • Inline, which is the default setting, displays the page element as an inline page element without a line break before or after the element. • Inline-block renders the page element as an inline rectangle, but with content that behaves as if it's inside a block element.
Position	When used in conjunction with the Absolute or Relative positioning options, these four properties place page elements outside the normal flow of the document. Enter a value in the text boxes as appropriate and select a unit of measurement in the respective drop-down lists. • Top sets how far the top edge of an element is above or below the top edge of the parent element. • Bottom determines how far the bottom edge of an element is above or below the bottom edge of the parent element. • Right sets how far the top edge of an element is to the right or left of the right edge of the parent element. • Left defines how far the left edge of an element is to the right or left of the left edge of the parent element.
Z-index	Specifies the order in which elements overlap each other when they need to be rendered in the same space. An element with a greater z-index value covers an element with a lower value. The default value is 0. Click  and  to increase and decrease the z-index, or enter a value in the text box.
Float	Floats a page element to the left or right so that subsequent elements—text for example—wrap around the floating page element. •  floats the page element to the left. •  floats the page element to the right. • **None** removes an existing float setting.

Property	Description
Clear	Specifies whether the selected page element allows floating page elements beside it. •  moves the page element below any floating page element on its left. •  moves the page element below any floating page element on its right. •  moves the page element below floating page elements on either side. • **None** removes existing float settings.
Visibility	Specifies whether the selected page element is visible. • Visible is the default value. • Hidden hides the page element and renders an invisible rectangle in its place. • Collapse is used to hide table elements. (For other page elements, it has the same result as hidden.) Note: Invisible page elements still occupy the same space in the page's layout.
Overflow	Specifies whether the content of a page element should be clipped when it overflows its area. • Visible does not clip the content. • Hidden clips the content. • Scroll clips the content, but provides scroll bars so that users can view the remaining content. • Auto is dependent on the browser, but should display a scroll bar to view the rest of the content.

The Dimensions Section

Property	Description
Width	Sets the width of the selected page element. Enter a value in the `Width` text box and select a unit of measurement. Select Inherit to use the width of the parent page element.
Height	Sets the height of the selected page element. Enter a value in the `Height` text box and select a unit of measurement. Select Inherit to use the height of the parent page element.
Margins	Sets the width of the page element's margin, which is the space between its border and outer edge. Set the margins for all four sides by entering a value in the `All` text box, or add margins to the top, right, bottom, or left sides as required.
Padding	Sets the width of the page element's padding, which is the space between its content and border. Set the padding for all four sides by entering a value in the `All` text box, or add padding to the top, right, bottom, or left sides as required.

The Borders Section

Property	Description
Type	Specifies whether to set border properties for each side separately or for all four sides.
Style	Sets the border's style such as dashed, dotted, or double.
Color	Sets the border's color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Color drop-down list.
Thickness	Specifies the border's thickness. Enter a value in the `Thickness` text box and select a unit of measurement. Alternatively, select Thin, Medium, or Thick.

The Tables Section

Property	Description
Border Collapse	When designing tables: • Collapse uses a common border between cells • Separate gives each cell its own border
Horizontal Spacing	Sets the horizontal distance that separates cell borders. Enter a value in the text box and select a unit of measurement. This value is used only if Border Collapse is set to Separate.
Vertical Spacing	Sets the vertical spacing that separates cell borders. Enter a value in the text box and select a unit of measurement. This value is only used if Border Collapse is set to Separate.

SEE ALSO:

Using the Style Pane

Understanding the Style Sheet View in Site.com

When working with style sheets, you can add style items, organize them into groups, and edit the CSS code directly. Open a style sheet on the Overview tab by double-clicking it or hovering over it and clicking > **Edit**. The style sheet opens as a new tab.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

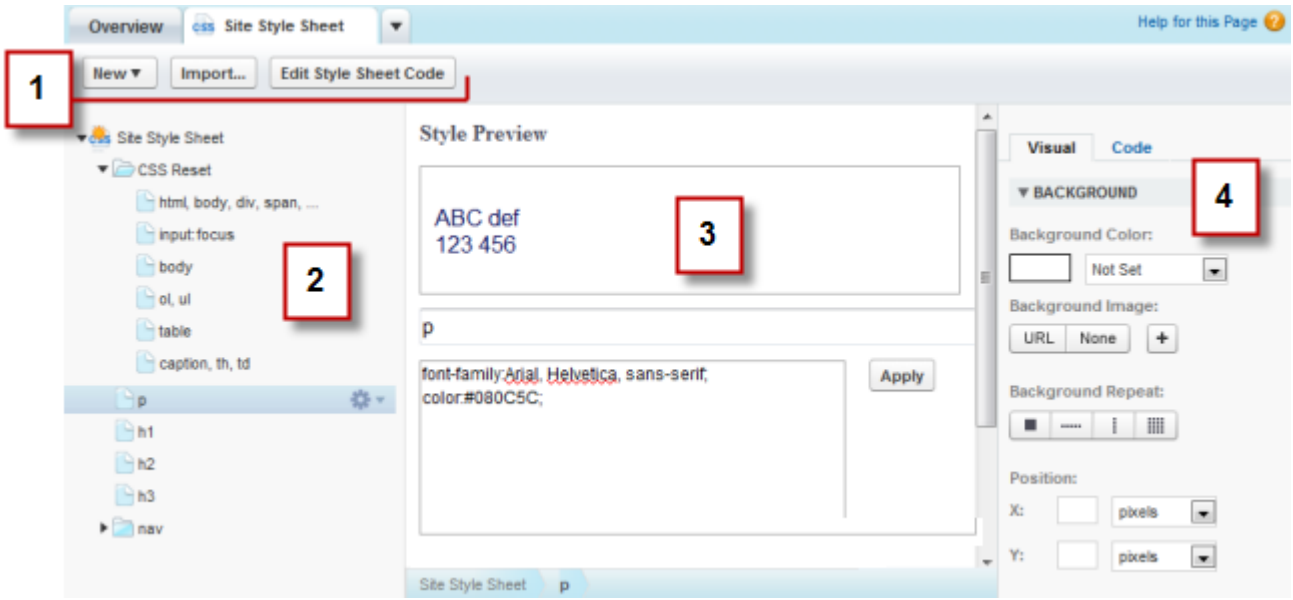
USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level



- Using the toolbar (1), you can import a style sheet and edit the style sheet's CSS code directly.
- Using the style sheet pane (2), you can:
  - Create style items and groups
  - Preview, edit, and delete style items

  - Move style items and groups by dragging them to the correct location
  - Add a CSS reset
- Using the Style Preview section (3), you can preview and manually edit a selected style item.

  Note: You can't preview at-rules, such as `@media` or `@font-face`, in the Style Preview section.
- Using the visual CSS editor (4), you can define the CSS properties for the selected style item.

SEE ALSO:

Creating and Using CSS Style Sheets

Cascading Style Sheets Overview

Creating and Using CSS Style Sheets

A default style sheet called "Site Style Sheet" is included with every site you create. However, if you're familiar with CSS and need multiple style sheets, you can create new ones to use in your site.

To create a style sheet:

1. Click **Style Sheets** > **New** on the Overview tab. Alternatively, click **New Style Sheet** in the Style Sheets view.
2. Enter a name for the style sheet.
3. Click **Apply**. The style sheet opens.
4. Add style items and groups to the style sheet.

Note:

- Style sheet names can only contain alphanumeric characters, hyphens, colons, and underscores.
- You can also import a CSS file to use in your site.

After you create a new style sheet, you must attach it to a page to apply its styles to the page.

To attach a style sheet to a page:

1. Select the page in the Page Structure pane ( ).
2. In the Style Sheets section of the Properties pane ( ), click +.
3. Select the style sheet in the list that appears.
4. Attach the style sheet to the page by clicking + beside the drop-down list.

Tip: If you used a page template to create your site pages, the quickest way to include the new style sheet on every pages is to attach it to the template. This automatically includes a reference to the style sheet in every page that's based on the template.

SEE ALSO:

Understanding the Style Sheet View in Site.com

Cascading Style Sheets Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating Style Sheet Items and Groups

When adding style items to style sheets, you can define CSS classes and IDs, or you can redefine the formatting of HTML tags such as `body` or `h1`. When you change the CSS style of an HTML tag, anything formatted with that tag is immediately updated.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Creating Style Items

To open a style sheet, double-click it in the Style Sheets view of the Overview tab, or hover over it and click > **Edit**.

If you're very familiar with CSS and prefer coding by hand, click **Edit Style Sheet Code** to edit the style sheet directly using the CSS editor. Additionally, to add at-rules (for example, `@media`), you must edit the style sheet directly.

Alternatively:

1. Select the style sheet and click > **Insert Style Item**.
2. Enter the name of the style item:
   - To redefine the default formatting of a specific HTML tag, enter the HTML tag name—for example, `body` or `h1`.
   - To create a CSS class, enter the class name and ensure that you include a period before it—for example, `.classname`.
   - To create a CSS ID, enter the ID name preceded by `#`—for example, `#contentID`.
3. Click **Apply**.
4. Add style definitions by either:
   - Setting style properties in the visual style editor on the right
   - Typing CSS styles in the text box in the Style Preview section and clicking **Save**

As you modify the definition of a selected style item, you can see how your changes appear in the Style Preview section.

Tip:

- A class name must begin with a period or it will not be recognized as a CSS class.
- An ID name must begin with `#` or it will not be recognized as a CSS ID.
- Use IDs when there is only one occurrence per page. Once you've used the ID, you can't use it again on that page. Use classes when there are one or more occurrences per page.
- Class and ID names can contain alphanumeric characters, hyphens, and underscores only, and can't begin with a number or include spaces.

Creating Style Groups

Use groups to organize your CSS logically. This makes it easier to locate and maintain styles.

When the style sheet is open:

1. Select the style sheet and click > **Insert Style Group**.
2. Enter a name for the group and click **Apply**.
3. To add a new style to the group, select the group and click > **Insert Style Item**. To add an existing style to the group, drag it onto the folder icon.

Assigning Style Items

After you've created styles, you can assign them to the pages and pages elements of your site.

To assign a class to a page or page element, select it and either:

- Type the class name in the `Class` field of the Properties pane ( ).
- Select **Class** in the Style pane ( ), start typing the name, and select it in the list that appears.

To assign an ID to a page or page element, either:

- Type the ID name in the `ID` field in the Properties pane ( ).
- Select **ID** in the Style pane ( ) and select it in the drop-down list.

SEE ALSO:

Creating and Using CSS Style Sheets
Understanding the Style Sheet View in Site.com
Cascading Style Sheets Overview

Using CSS Reset

Every browser has set presentation defaults, but unfortunately they aren't standardized across all browser types. This means that when you use CSS to style your site, it may not render as expected when you view it in different browsers. For example, browsers differ in how they display:

- Unordered and ordered lists
- Top and bottom margins for headings
- Indentation distances
- Default line-heights

A CSS reset cancels the differences between browsers to control how browser elements are presented to the end user. You can either use Site.com's CSS reset, or you can add your own CSS reset code.

To use Site.com's CSS reset:

1. In the Style Sheets view on the Overview tab, open the style sheet by double-clicking it, or hovering over it and clicking > **Edit**.
2. Click > **Insert CSS Reset**.
3. Ensure the CSS reset is positioned at the top of the style sheet. To move it, drag it to the correct location in the pane on the left.

To add your own CSS reset code:

1. In the Style Sheets view on the Overview tab, open the style sheet by double-clicking it, or hovering over it and clicking > **Edit**.
2. Click **Edit Style Sheet Code** to open the CSS editor.
3. Paste the code at the top of the style sheet code.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

4. Click **Save and Close**.

SEE ALSO:

- Creating and Using CSS Style Sheets
- Cascading Style Sheets Overview

Branding Your Site

Site Branding Overview

Branding provides a flexible way for you to define different aspects of your website's brand. Once branding properties are defined, your editors can easily customize everything in one centralized place, the Branding Editor. When your website editors customize the properties, they get a preview of their branding changes immediately.

By adding branding properties to your website, page templates, or widgets, you can easily define and change those aspects of your website's brand without needing to edit your Cascading Style Sheet.

Here are examples of aspects that you might define that are related to your website's brand:

- The website's background, page, or link colors
- Font size and color
- The logo or header background image
- The thickness and color of lines

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

To use branding in your website, you must:

1. Define the parts of your site that make up your brand by creating a branding property for each aspect such as background color, logo, and fonts. You complete this task from Branding Properties within Site Configuration.
2. Use expression language syntax to replace the current definitions for these properties within your Cascading Style Sheets with the new branding properties.
3. Use the Branding Editor to customize the properties and preview how your website will look.

The branding properties can also be accessed by using expression language syntax directly from within a custom code block or content block.

SEE ALSO:

- Creating Branding Properties
- Setting Up Branding Properties
- Using the Branding Editor

Creating Branding Properties

With branding , you can quickly create style properties that can be reused by editors of your site.

When creating your branding properties, you can organize properties into multiple sections to make them easier to find in the Branding Editor. You can also organize your properties within sections.

- Order properties within sections in a logical manner. For example, organize them alphabetically.
- Order sections and properties by dragging and dropping them within the Branding Properties view.

1. From within your site, click **Site Configuration** > **Branding Properties**.
   The Branding Properties editor appears.
2. Click 

.
3. Enter a name for the property in the Label field.
   The expression name is filled in automatically. The expression name is used in style sheets and code blocks.
4. Choose a type.
5. Set the default value.
6. To make the property required, click **Required**.

Hover over any property and use the menu  to edit or delete it. You can double-click any section name to edit it.

SEE ALSO:

- Site Branding Overview
- Custom Property Types
- Setting Up Branding Properties
- Using the Branding Editor

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Setting Up Branding Properties

After you create branding properties, the next step is to reference them as expressions in your Cascading Style Sheets (CSS) or Site.com page elements such as content blocks or custom code elements.

The expression syntax for branding properties is `{!Site.branding-property-name}`. For example, if you create a color property with the name HeaderColor, you can insert the property into a CSS or code block by typing `{!Site.HeaderColor}`.

Note: The expression names are case-sensitive!

1. Open your CSS or code block.
2. Locate the place where you want to insert the expression.
3. Type `{!`
   and a drop-down list of available properties appears.
4. Select the property and double-click to enter it on the page.

Example: For example, if part of your website's brand is to use blue for the background of the header section,you would normally statically set the color's HEX value in your CSS rules.

```
.site-header {
 background-color: #3793DD;
}
```

With branding, you can create a property that maps directly to the HEX color and then use the property's expression name in your CSS.

```
.site-header {
 background-color: {!Site.HeaderColor};
}
```

Now the color can be easily changed by any of your site editors by using the Branding Editor instead of editing the CSS.

SEE ALSO:

- Site Branding Overview
- Creating Branding Properties
- Using the Branding Editor

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Using the Branding Editor

The Branding Editor provides a centralized place to customize and then preview the changes that you make to the branding properties in your site.

The Branding Editor includes two areas:

- The editor palette, on the left side, contains all of the branding properties that you can edit.
- The preview area on the right side shows a live preview of how your website will appear after you change a style.

1. Open the page that you want to edit.
2. Click  .
3. From the left pane, select the style property that you want to edit.
   Your results appear immediately in the right pane.

SEE ALSO:

Site Branding Overview

Creating Branding Properties

Setting Up Branding Properties

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Using Custom Properties

Custom Site Properties Overview

With custom site properties, you can define and store frequently occurring content on your site. For example, you can store your address and phone number as a custom property so that it can be reused by anyone who is editing your site. You can apply stored properties to pages, headers and footers, and widgets quickly by using expression language syntax.

To use custom properties in your website, you must:

1. Define the custom property. You complete this task from Custom Properties within Site Configuration.
2. Use expression language syntax to replace the current definitions for these properties within your Cascading Style Sheets or custom code blocks.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

After you create a new property and specify its value and type, you can access the property in custom code or content blocks by using expressions. Expressions are placeholders for data that will be replaced with the custom property when the page loads.

SEE ALSO:

- Creating Custom Site Properties
- Setting Up Custom Properties

Creating Custom Site Properties

You can quickly create Custom Properties that can be reused by editors of your site.

When creating your custom properties, you'll probably want to organize them logically.

- Organize properties into multiple sections to make them easier to manage and easy to find.
- Order properties within sections in a logical manner. For example, organize them alphabetically.
- Order sections and properties by dragging and dropping them within the Custom Properties view.

1. From within your site, click **Site Configuration** > **Custom Properties**.
   The Custom Properties view appears.
2. Click + New Property.
3. Enter a name for the property in the Label field.
   The expression name is filled in automatically. The expression name is used in style sheets and code blocks.
4. Choose a type.
5. Set the default value.
6. To make the property required, click **Required**.

To edit or delete properties, hover over any property and use the menu   . To edit section names, double-click the name make changes.

SEE ALSO:

- Custom Site Properties Overview
- Custom Property Types
- Setting Up Custom Properties

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Setting Up Custom Properties

You can use custom properties in your website by using expression language syntax within a style sheet, code block, widget, or the HTML page editor.

To reference a site custom property in custom code or a content block, use the syntax `{!Site.expression_name}`. For example, create a custom property to store the company phone number with PhoneNumber as the expression name. In a content block, enter `Contact us at {!Site.PhoneNumber}`. When the page loads, it replaces the value that is represented by {!Site.PhoneNumber} and displays "Contact us at 1-800-667-6389" on the page.

Note: The expression names are case-sensitive!

1. Open a website element such as a code block or widget.
2. Locate the place where you want to insert the expression.
3. Type `{!`
   to see the list of available custom properties.
4. Select the property and double-click to enter it on the page.

Example: Setting up custom properties is great for things that might change over time such as your address or a particular product phrase. Because expressions are just placeholders, if you update the value of a property in the Custom Properties view, the value is updated automatically wherever the custom property is referenced on a page.

SEE ALSO:

Custom Site Properties Overview

Creating Custom Site Properties

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Custom Properties for Page Templates or Widgets Overview

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property to achieve greater flexibility over how templates and widgets are reused.

Then, by adding custom code or content blocks to the page template or the widget, you can access the property values by using expressions. Expressions serve as placeholders for data that will be replaced with information when the page loads.

In turn, when you or your team create a page from the template, the page is a copy or *instance* of the template. Similarly, when you add the widget to a page, it creates an instance of the widget. You can't edit the instance, but you can update its property values.

Because expressions are just placeholders, their values are updated automatically when you update the values in the Properties pane of the page or widget.

You can also create sections to group related properties. These sections control how properties are grouped in the Properties pane.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

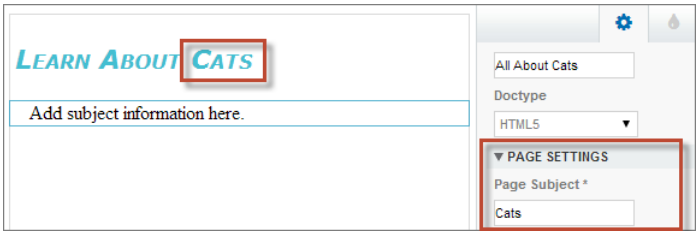
Available (with limitations) in: **Developer** Edition

Example: For example, let's say you add a content block to a template to contain the page's heading. In this case, when users create a page from the template, you want to let them replace part of the text to suit their needs, but without letting them edit the entire content block.

By adding a custom property that's called `pageSubject` and specifying an initial value, you can instead use the following expression in the content block:

```
Learn About {!pageSubject}
```

This action lets team members rename any page that's derived from the template by updating the `Page Subject` property in the page's Properties pane, which automatically updates the value that's represented by the `{!pageSubject}` expression.



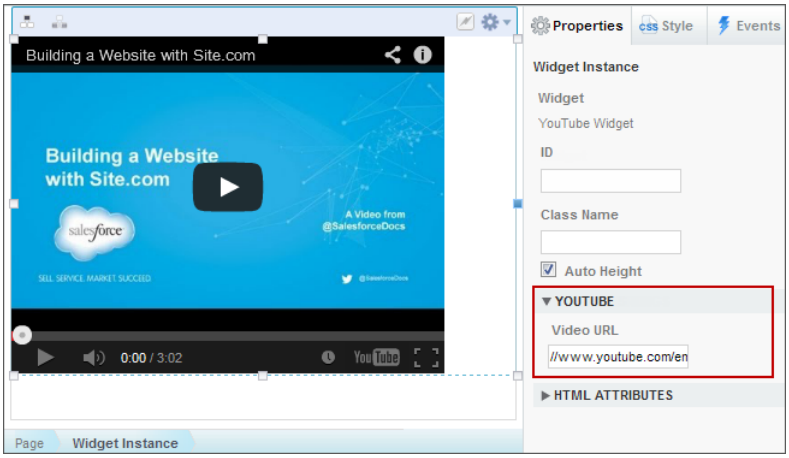
Example: Alternatively, let's say you want to create a YouTube widget using the following embed code:

```
<iframe width="560" height="315" src="//www.youtube.com/embed/hcUaN6XBTz4"
frameborder="0" allowfullscreen></iframe>
```

However, you want users to specify which video to display when they add an instance of the widget to the page. In this case, you could create a section called *YouTube*, add a custom property labeled *Video URL* with *videoURL* as the expression name, and instead use the following code:

```
<iframe width="560" height="315" src="{!videoURL}" frameborder="0"
allowfullscreen></iframe>
```

Now, when users add the YouTube widget to the page, they can point to any video by updating the *Video URL* property in the YouTube section of the Properties pane, which automatically updates the value represented by the `{!videoURL}` expression.



SEE ALSO:

- Widgets Overview
- Adding Custom Properties to Page Templates or Widgets
- Custom Property Types
- About Displaying Dynamic Data Using Expressions

Adding Custom Properties to Page Templates or Widgets

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property.

Then, by adding custom code or content blocks to the page template or widget, you can access the property values using expressions, which serve as placeholders for the values. When you or your team create a page from the template or add an instance of the widget to the page, you can update the property values to modify the instance.

When the page template or widget is open:

1. Make sure the template or widget is selected in the Page Structure pane and click **Edit Custom Properties** in the Properties pane.
2. Click **New Property**.

   Base is the default section, which you can rename. Sections control how properties are grouped in the Properties pane of a widget instance or page.
3. Enter the property label, which appears in the Properties pane of a widget instance or page—for example, *Video URL*.
4. Optionally, update the expression name that was added automatically.

   Expression names are case sensitive and can't include spaces.
5. Specify the property type and value.
6. Optionally, mark the property as required.
7. Add additional properties or sections as needed. Reorder items by dragging them to a new location.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

8. To use a custom property, drag a **Custom Code** or a **Content Block** page element from the Page Element tab onto the page template or widget.
9. Use the syntax `{!expression_name}` to reference the expression—for example, `{!videoURL}`. When the page loads, the expression will be replaced by the property value.
10. Add any additional text you require and save your changes. For example, `Check out our video at {!videoURL}`.

Tip: You can also create a custom property by first typing the name of the property in custom code or a content block using expression language. For example, if you type `{!videoHeight}` in a content block, a new property called `videoHeight` is automatically added to the Properties pane of the template or widget, where you can then add a property value.

If you update a widget's properties, your changes won't be reflected in any widget instances. Instead, you must replace existing widget instances with the latest version.

If you delete a custom property that's being used in custom code or a content block, make sure you remove any references to the property.

SEE ALSO:

- Custom Properties for Page Templates or Widgets Overview
- Custom Property Types
- About Displaying Dynamic Data Using Expressions
- Creating Widgets

Custom Property Types

Property types are available when designers or site administrators create site branding properties, site custom properties, widget branding properties, and widget custom properties.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Property Type	Description
`Text`	Lets users enter a text value.
`Checkbox`	Lets users set a true (selected) or false (deselected) value.
`Color`	Lets users specify a color, either by selecting one from the color picker or by entering a hexadecimal value, such as `#333333`.
`Image`	Lets users specify an image, either by browsing to a site image, uploading a new image, or entering the URL to an image.
`HTML`	Lets users enter an HTML value.
`Units`	Lets users enter a measurement, such as pixels, percentage, or em—for example, `5px`.
`Picklist`	Lets users select a value from a list you define. To add items to the picklist, create a comma-delimited list of options, where each option has a label and a value—for example, `Label 1:Value 1, Label 2:Value 2`. If you don't add a value for an option, the label is used as the value by default.

Property Type	Description
	To define a default picklist selection, use the format `label:value:default`—for example, `Small:S:default, Large:L`.
`Font`	Lets users select a font.

SEE ALSO:

- Adding Custom Properties to Page Templates or Widgets
- Creating Custom Site Properties
- Creating Branding Properties

Working with Dynamic Data

Site.com Data Services Overview

Site.com data services combine many features that let you connect to standard and custom Salesforce objects. Retrieve data from your organization's objects and dynamically display it on your site pages, or alternatively, gather and submit data from your customers. And when you update data in your Salesforce object, the changes are reflected automatically on the live site—no site updates required!

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Here are a few ways you can use Site.com data services:

- Publish a catalog of products—List your company's products and include information, such as model numbers and prices, pulled dynamically from your organization.
- Post company press releases—Publish your company's press releases and sort by publication date.
- Create a realtor website—Display current listings filtered by city or price.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes.

So how does it all work? Several data-bound page elements let you retrieve and display your data, or collect data from your site visitors.

Data tables connect to Salesforce objects, retrieve a data set based on the filter criteria that you specify, and display one or more record as rows in the table.

Data Repeaters and data elements combine to let you connect to standard and custom objects, retrieve data, and dynamically display it on your site's pages. Together, the data repeater and data elements result in a "repeating template" that offers you the greatest flexibility for displaying one or more records on the page.

Data functions let you perform calculations on data retrieved from objects and display the result on the page. For example, for a particular field in an object, you can use a data function to calculate the total value or the average amount of all returned records.

Nested repeaters let you retrieve data from objects with a parent-to-child relationship.

Forms and form fields combine to let you collect data from your site visitors and submit the data to standard or custom Salesforce objects. Create web-to-lead forms, capture customer details, or gather feedback on your products or services.

Data Services Considerations

- To allow guest users to view the data in or submit data to a Salesforce object, you must set the object's data access permissions.

- When working with assets, the easiest way to take advantage of Site.com data services is to import the files into your website, and store a relative URL to these assets in your standard or custom object. See Storing Assets to Use with Salesforce Objects on page 747.
- If you add a data-bound page element to your site and then subsequently change a field type in the Salesforce object it's connected to—for example, changing a text field to a picklist—the data-bound page element will no longer work. You must reconfigure the data-bound page element to reference the updated field.
- If you update data in an object that's connected to a data table, data repeater, or data function, the changes are reflected automatically on the live site. To control this, you can add a picklist field to the object to specify when a record is approved to go live. Then you can use the field to filter the records by approved status, so only approved records appear on the live site.

SEE ALSO:

Accessing Data in Related Objects Overview
Adding Pagination to Data Repeaters and Data Tables
The Default, Error, and No Data Views

Setting Data Access Permissions for Salesforce Objects

Sites built with Site.com are publicly available, so visitors access the site via the Guest User license that's associated with the site.

By default, site visitors can access information made available in an active public site, such as the site's pages and assets. However, to allow guest users to view or submit data to a Salesforce object, you must modify the object's permission in the site's guest user profile. Each site has a separate Guest User license, so you can control guest access to Salesforce objects on a per-site basis.

To edit the site's guest user profile:

1. On the Overview tab of Site.com Studio, click **Site Configuration** and click *Site Name* **Profile**. Alternatively, if you're adding a data repeater, data table, data function, or form to the page, click **go to the guest user profile** in the item's dialog box.
2. In the site's guest user profile, enable the "Read" permission on the standard or custom objects you want to retrieve data from using data repeaters, data tables, or data functions. Enable the "Create" permission on the objects you want to submit data to using forms. All permissions that aren't set by default must be set manually.
3. If required, modify the field-level security of an object.

SEE ALSO:

Storing Assets to Use with Salesforce Objects

Site.com Data Services Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

Storing Assets to Use with Salesforce Objects

Because websites built with Site.com are publicly available, site visitors don't have the security privileges required to view images and documents stored in your Salesforce objects, which are available to authenticated users only.

Therefore, the easiest way to take advantage of Site.com data services is to import the files into Site.com, and instead store a relative URL to these assets in your standard or custom object. Alternatively, if your images or files are hosted elsewhere and readily available on the Internet, you can use an absolute URL.

For example, let's say you want to use a custom object called "Catalog" to build a Catalog site page that displays product details and images. Before you begin building the Catalog site page, you would:

1. Import the product images into your Site.com site.
2. In the Catalog object, create a field to store the relative URL of the image, such as `Image URL`.
3. For each product record, add the relative path of the image. This URL is relative to the site, so if you upload `widget.png` to your site, the relative path is `/widget.png`. URLs are case sensitive.

Then, when you add a data table, or a data repeater and data elements to the Catalog site page to display the product data, you can reference the `Image URL` field to dynamically display each product's image on the page.

SEE ALSO:

- Setting Data Access Permissions for Salesforce Objects
- Site.com Data Services Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Dynamically Retrieving and Displaying Data

Dynamically Retrieving Data with Data Repeaters

Use a data repeater to connect to a standard or custom Salesforce object and retrieve a data set based on the filter criteria that you specify. When you combine a data repeater with data elements, custom code, or content blocks, you can create a "repeating template" that displays one or more records on the page.

To add a data repeater to a page:

1. Drag a **Data Repeater** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.

   Note:

   - For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's "Read" permission, and refresh the list.
   - For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
   a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *`parent_object_name.field_name`*.
   b. Select the operator to control how results are filtered. For example, select *`Equals`* to return an exact match.
   c. Select the source of the filter value. For example, to specify an explicit value, select *`Fixed value`*, or to use the values passed to the page via a query string, select *`URL query string`*.
   d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
   e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

   Note: If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

4. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
5. In Limits, you can limit the number of returned results. For example, if you're only interested in the top five results, enter *`5`* in the `Limit results to` field.
6. If you're adding pagination, specify the number of results to display per page in the `Results per page` field.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

7. Click **Save**.

Next, you must add either data elements, custom code, or content blocks to the data repeater to display the data it retrieves.

SEE ALSO:

- The Default, Error, and No Data Views
- Improving Performance Using Caching
- Data Filtering Examples
- Site.com Data Services Overview

Displaying Data Using Data Elements

You can use a data element to display the data retrieved by a page data connection or a data repeater. The data element binds to a field in the object and acts as a placeholder that's replaced with the field's data when the page loads.

When combined with a data repeater, data elements result in a "repeating template" that displays one or more records on the page. When used with a page data connection, data elements display data from a single record.

You can use data elements to display plain text, formatted text (for dates and numbers), or images. You can also add hyperlinks to data elements to allow site visitors to navigate to another page, such as a detailed description, or to refresh the data displayed on the page or the data repeater based on their selection. See Data Filtering Examples on page 768.

When the page is open:

1. Drag a **Data Element** from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Data Element** page element directly onto the page canvas.
2. Select the field to display. To customize how the field's data is displayed, click **Customize**.

   Note: The object's fields are listed first, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.

3. Select the display type.

Option	Description
Text	Lets you display the field's data as plain text.
Formatted text	Lets you choose from several text display formats if you're working with dates, times, or currency.
Image	Lets you display the field's data as an image if the field contains an image URL. The URL can be absolute or relative to the site. You can also select a field to use for the alternative text or enter custom text.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

4. To create a hyperlink, select `Add a hyperlink`. Otherwise, go to step 8.
5. Select the link type.

Option	Description
A URL	Lets you link to a Web page by: • Choosing a field that you want to reference, such as a field that stores the relative URLs of PDFs you uploaded to your site. • Choosing a field that you want to reference and clicking **Customize** to add an absolute URL or to create a custom link, such as a URL query string.
An item in your site	Lets you link to a page, image, or file in the site by selecting the item type and then selecting the item. (If you can't see the list of items, place your cursor in the `URL` field and press the DOWN key on your keyboard.) You can also customize the URL—for example, by creating a URL query string.
An email	Lets you link to an email message by entering the recipient's address, and the message subject and body. You can use merge fields to access the object's fields. For example, if an object has an `Email` field, enter the merge field, such as `{!email}`, in the Email address text box. When the link is clicked, it opens a new message window in the user's email client and adds the appropriate email address to the To: field.

6. Optionally, enter a tooltip by selecting the required field or clicking **Customize** to add custom text.
   The tooltip displays as a pop-up when the user hovers over the link.
7. If you're linking to a URL or an item in your site, specify where the item should open.

Option	Description
Popup window	Loads the item into a popup window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
New window (_blank)	Loads the item into a new, unnamed browser window.
Same window (_self)	Loads the item into the same frame or window as the link. This is the default setting.
Topmost window (_top)	Loads the item into the topmost parent frameset or window of the frame that contains the link.

Option	Description
Parent window (_parent)	Loads the item into the parent frameset or window of the frame that contains the link.

8. Click **Save**.

The data element is displayed on the page as a merge field. To test the output, preview the page.

SEE ALSO:

- Dynamically Retrieving Data with Data Repeaters
- Retrieving Data with Page Data Connections
- Displaying Data Using Custom Code
- Displaying Data Using Content Blocks

Displaying Data Using Custom Code

In addition to data elements and content blocks, you can also use custom code as an alternative way to display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

When the page is open:

1. Drag a **Custom Code** page element from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Custom Code** page element directly onto the page canvas.
2. To access the fields of the object that the data repeater or page is connected to, type `{!` and double-click the expression that you want to display.

   Note: The object's fields are listed first, followed by the fields of all parent objects, which use the format *`parent_object_name.field_name`*.

3. Add any additional expressions or text you require. For example:

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

```
To contact {!Name}, call {!Phone}.
```

where `{!Name}` and `{!Phone}` are placeholders for the values of the `Name` and `Phone` fields of each record.

4. Click **Save and Close**.

SEE ALSO:

- Dynamically Retrieving Data with Data Repeaters
- Retrieving Data with Page Data Connections
- Displaying Data Using Content Blocks
- Data Filtering Examples

Displaying Data Using Content Blocks

In addition to data elements and custom code, you can also use content blocks as an alternative way to display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

When the page is open:

1. Drag a **Content Block** page element from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Content Block** page element directly onto the page canvas.
2. Enter the name of the field you want to display using an expression. For example, `{!Name}`.

   Note: The object's fields are listed first, followed by the fields of all parent objects, which use the format *`parent_object_name.field_name`*.

3. Add any additional expressions or text you require. For example:

   ```
   To contact {!Name}, call {!Phone}.
   ```

   where *`{!Name}`* and *`{!Phone}`* are placeholders for the values of the `Name` and `Phone` fields of each record.

   You can also use expressions if you're adding a hyperlink to the content block.

4. Click **Save**.

SEE ALSO:

- Dynamically Retrieving Data with Data Repeaters
- Retrieving Data with Page Data Connections
- Displaying Data Using Custom Code
- Displaying Data Using Data Elements
- Data Filtering Examples

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Dynamically Retrieving Data with Data Tables

Use a data table to connect to a standard or custom Salesforce object, retrieve a data set based on the filter criteria that you specify, and display one or more record as rows in the table. A data table's columns bind to the fields of the object it's connected to. Each column cell acts as a placeholder that is replaced with the field's data when the page loads.

To add a data table to the page:

1. Drag a **Data Table** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.

   Note:

   - For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's "Read" permission, and refresh the list.
   - For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
   a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
   b. Select the operator to control how results are filtered. For example, select *Equals* to return an exact match.
   c. Select the source of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
   d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
   e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

   Note: If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

4. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
5. In Limits, you can limit the number of returned results. For example, if you're only interested in the top five results, enter *5* in the `Limit results to` field.
6. If you're adding pagination, specify the number of results to display per page in the `Results per page` field.
7. Click **Next**.
8. Add available fields to the table by double-clicking a field, or selecting it and clicking >.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

9. Reorder the list of selected fields by clicking **Move Up** or **Move Down**.
10. Click **Save**.

Note: You can't add page elements to a data table. However, you can add additional columns to a data table by selecting it and clicking **Edit**. On the Select Fields screen, select the additional fields and save your changes.

After you've added the data table to the page, you can use the Properties pane to:

- Provide a short heading or summary of the table's purpose in the `Caption` field. The caption appears above the table, and complies with W3C accessibility standards for screen reader users.
- Hide the column headings by deselecting `Show Column Headings` in the Table section.
- Make the columns sortable by selecting the jQuery Flexigrid theme in the Theme section and selecting `Enable Sorting`. The theme also changes the appearance of the table.
- Change the name of a column by selecting the column cell and updating its name in the `Column Heading` property in the Data Table section.

SEE ALSO:

Editing Columns in a Data Table
The Default, Error, and No Data Views
Improving Performance Using Caching
Data Filtering Examples

Editing Columns in a Data Table

A data table's columns bind to the fields of the object it's connected to. Each column cell acts as a placeholder that is replaced with the field's data when the page loads.

You can display plain text, formatted text (for dates and numbers), or images in the column cells. You can also add hyperlinks to column cells to allow site visitors to navigate to another page, such as a detailed description, or to refresh the data displayed in the data table based on their selection. See Data Filtering Examples on page 768.

To edit a column:

1. Double-click the column cell in the data table.
2. Select the field to display. To customize how the field's data is displayed, click **Customize**.

   Note: The object's fields are listed first, followed by the fields of all parent objects, which use the format *`parent_object_name.field_name`*.

3. Select the display type.

Option	Description
Text	Lets you display the field's data as plain text.
Formatted text	Lets you choose from several text display formats if you're working with dates, times, or currency.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Option	Description
Image	Lets you display the field's data as an image if the field contains an image URL. The URL can be absolute or relative to the site. You can also select a field to use for the alternative text or enter custom text.

4. To create a hyperlink, select `Add a hyperlink`. Otherwise, go to step 8.
5. Select the link type.

Option	Description
A URL	Lets you link to a Web page by: • Choosing a field that you want to reference, such as a field that stores the relative URLs of PDFs you uploaded to your site. • Choosing a field that you want to reference and clicking **Customize** to add an absolute URL or to create a custom link, such as a URL query string.
An item in your site	Lets you link to a page, image, or file in the site by selecting the item type and then selecting the item. (If you can't see the list of items, place your cursor in the `URL` field and press the DOWN key on your keyboard.) You can also customize the URL—for example, by creating a URL query string.
An email	Lets you link to an email message by entering the recipient's address, and the message subject and body. You can use merge fields to access the object's fields. For example, if an object has an `Email` field, enter the merge field, such as `{!email}`, in the Email address text box. When the link is clicked, it opens a new message window in the user's email client and adds the appropriate email address to the To: field.

6. Optionally, enter a tooltip by selecting the required field or clicking **Customize** to add custom text.
   The tooltip displays as a pop-up when the user hovers over the link.
7. If you're linking to a URL or an item in your site, specify where the item should open.

Option	Description
Popup window	Loads the item into a popup window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
New window (_blank)	Loads the item into a new, unnamed browser window.
Same window (_self)	Loads the item into the same frame or window as the link. This is the default setting.
Topmost window (_top)	Loads the item into the topmost parent frameset or window of the frame that contains the link.
Parent window (_parent)	Loads the item into the parent frameset or window of the frame that contains the link.

8. Click **Save**.

The column is displayed on the page as an expression. To test the output, preview the page.

To change the name of a column, select the column cell and update the name in the `Column Heading` field of the Properties pane.

SEE ALSO:

Dynamically Retrieving Data with Data Tables

Site.com Data Services Overview

Adding Pagination to Data Repeaters and Data Tables

Events let you add interactive and animated effects to the pages and page elements of your website. When using data repeaters and data tables, you can add pagination events so users can easily page through the displayed data. This is particularly useful when working with large amounts of data.

For example, if you've added a data repeater that displays all the users in an organization, you can add pagination to help users navigate through the data. You can add three pagination events:

- Previous Page
- Next Page
- Go To Page

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Creating Previous and Next Pagination

You can create previous and next buttons so users can move through the data one page at a time. The process for creating both buttons is the same.

1. Create your data repeater or data table.
2. In the data repeater or data table, be sure to specify how many records to display per page in the Limits section.
3. Drag a button to the page.
4. In the Properties pane, change the Button Name to Previous Page or Next Page as appropriate.
5. In the Events pane, select the click event.
6. When the Actions box appears, click + and select the Previous Page or Next Page action.
7. In the Target Element, select the data repeater or data table.
8. Click **Save**.

Creating GoTo Pagination

Creating GoTo navigation is similar to creating the previous and next buttons, but you must add an input field so users can specify what page they what to go to.

1. Create your data repeater or data table.
2. In the data repeater or data table, be sure to specify how many records to display per page in the Limits section.
3. Drag a **Number** field onto the page.
4. In the Properties pane, change the field's Label Name to something that makes sense. For example, Enter Page Number.
5. Drag a **Button** onto the page.
6. In the Properties pane, change the Button Name to GoTo Page.
7. In the Events pane, select the click event.
8. When the Actions box appears, click + and select the Go To Page action.
9. In the Target Element, select the data repeater or data table.
10. For Input Field ID, select the field you created in step 3.

11. Click **Save**.

SEE ALSO:

Dynamically Retrieving Data with Data Repeaters
Dynamically Retrieving Data with Data Tables
Creating an Event
Available Events and Actions

Using Data Functions

A data function lets you connect to a standard or custom Salesforce object, perform calculations on the returned results, and display the calculation on the page. For example, for a particular field in an object, you can use a data function to calculate the total value or the average amount of all returned records.

To add a data function to a page:

1. Drag a **Data Function** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.

   Note:

   - For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's "Read" permission, and refresh the list.
   - For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
   a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
   b. Select the operator to control how results are filtered. For example, select *Equals* to return an exact match.
   c. Select the source of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
   d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
   e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

   Note: If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

4. In Functions, select a function:

Option	Description
Count	Counts the number of records that contain a value for the selected field. For example, if an object contains 30 records, but only 25 records have a value in the field you specify, the result is 25.
Maximum	Returns the highest value of all the values for the selected field. Applies to numbers, strings, and dates.
Average	Calculates the average value of all records for the selected field. For example, if you have 20 records with a total value of $20,000 in the Price field, the average is $1,000. Only applicable to fields that contain numbers.
Minimum	Returns the lowest value of all the values for the selected field. Applies to numbers, strings, and dates.
Sum	Calculates the total value of all records for the selected field.

5. Select the field it applies to.
6. Click **Save**.

SEE ALSO:

The Default, Error, and No Data Views
Improving Performance Using Caching
Data Filtering Examples

Page Data Connections Overview

Page data connections make it easy for site administrators and designers to create a detail page for a single record when working with Salesforce objects.

When combined with repeater elements, custom code, or content blocks, page data connections let you connect to standard and custom objects, retrieve a specific record, and dynamically display the record on a site page.

Example: For example, let's say you want to list all of your company's accounts on a site page called Accounts. When a user clicks a **View Details** link, you want to open a site page called Account Detail to display information for the selected account.

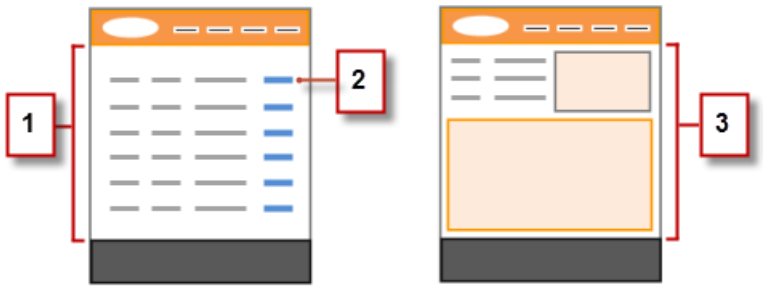
In this case, you could add a data repeater to the Accounts site page to retrieve a list of records from the Account object (1). Using a data element, you could create a **View Details** link (2) that, when clicked, uses a URL query string to pass the `Account ID` field as a unique identifier to the Account Detail page. As the Account Details page loads, the page data connection uses the unique identifier value to dynamically return only that record's details (3).

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition



With page data connections, you can use expressions to access the returned data anywhere on the page, including the page's properties. For example, let's say you want to use the account name as the title of the Account Detail page. In this case, you would simply enter `{!Name}` in the `Title` field on the Properties pane. When the page loads, it retrieves the account name for that particular record and displays it in the browser's title bar.

SEE ALSO:

Retrieving Data with Page Data Connections

Site.com Data Services Overview

Retrieving Data with Page Data Connections

Use a page data connection to create a detail page for a single record when working with Salesforce objects.

When combined with data elements, custom code, or content blocks, page data connections let you connect to standard and custom objects, retrieve a specific record, and dynamically display the returned data anywhere on the page. You can even use expressions to access the returned data in the page's properties.

When the page is open:

1. Ensure that the page is selected in the Page Structure pane.
2. Click **Add Connection** in the Page Data Connection section of the Properties pane.
3. Select the object that you want to connect to.

   Note:

   - For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's "Read" permission, and refresh the list.
   - For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

4. Optionally, in Filters, select criteria to filter which record is returned. If you don't select any criteria, the first record is returned by default.
   a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format `parent_object_name.field_name`.
   b. Select the operator to control how results are filtered. For example, select `Equals` to return an exact match.
   c. Select the source of the filter value. For example, to specify an explicit value, select `Fixed value`, or to use the values passed to the page via a query string, select `URL query string`.
   d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
   e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

   Note: If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

5. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order.
   For example, say you want to create a page that displays the most popular movie. You could connect to a custom object that contains movie data, and instead of using filter criteria, you could sort by user rating in descending order. Because a page data connection returns a single record, only the highest rated movie is returned.
6. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

Next, you must add either data elements, custom code, or content blocks to the page to display the retrieved record data.

You can use expressions to access the returned data anywhere on the page, including the page's properties. For example, let's say you want to use the account name as the title of the Account Detail page. In this case, you would simply enter `{!Name}` in the `Title` field on the Properties pane. When the page loads, it retrieves the account name for that particular record and displays it in the browser's title bar.

SEE ALSO:

- Page Data Connections Overview
- Improving Performance Using Caching
- About Displaying Dynamic Data Using Expressions
- Data Filtering Examples

Accessing Data in Related Objects Overview

Standard and custom objects have relationships that define how records in one object relate to records in another. For example, the Accounts object has a one-to-many relationship with the Contacts object—that is, each account can have one or more contacts associated with it. This relationship is also known as a parent-to-child or a master-detail relationship.

See Object Relationships for more information about relationships between objects in Salesforce.

Data repeaters, data tables, and data functions take advantage of these relationships to let you display data from related objects on the page.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

About Accessing Data in Parent Objects

When you add a data repeater, data table, or data function to the page and connect it to a standard or custom object, you can automatically access the fields of any parent object it's related to.

If you add filter criteria to a data repeater, data table, or data function, the Field drop-down list in the Create Data Connection dialog box displays the object's fields, followed by the fields of all of its parent objects, which use the format *`parent_object_name.field_name`*. This lets you filter results based on a field in the parent object. So for example, when retrieving records from the Contacts object, you could decide to return only contacts where the account name (`Account.Account Name`) is "ABC Labs."

Similarly, when you add data elements to a data repeater, or columns to a data table, you can bind them to fields in a parent object. So for example, if you add a data table that's connected to the Contact object, you can add a column that binds to its `Full Name` field and a column that binds to the Account object's `Account.Account Name` field to display the contact's name along with the name of the account it's associated with.

About Accessing Data in Child Objects

You can retrieve data from any child object of a parent object using a data repeater that contains another data repeater, data table, or data function. The outer or parent data repeater connects to an object, such as Accounts. In turn, the inner data repeater, data table, or data function automatically lets you connect to any child objects, such as Contacts. This is also known as a *nested data repeater*.

Let's say you want to display a list of accounts along with the names of the associated contacts, similar to this example.

ABC Labs

Francis Buchner

Acme Inc

Edward Stamos

Leanne Tomlin

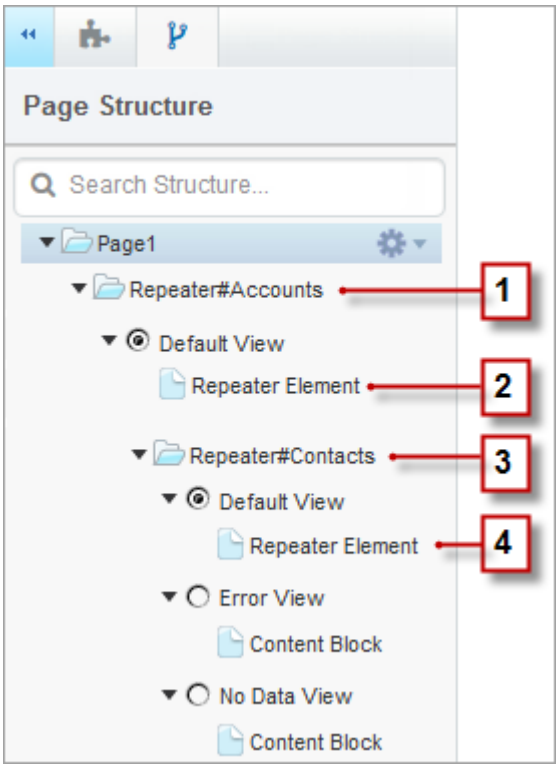
Dixon Chemical

Darrin Clark

Andy Rother

Jack Fallon

You can achieve this by creating a data repeater (1) that's connected to Accounts, and adding a data element (2) to it that binds to the `Account Name` field. Then add a nested data repeater (3) that's connected to Contacts, which is a child of Accounts. Finally, add a data element (4) to the nested data repeater that binds to the Contact object's `Full Name` field.



SEE ALSO:

- Displaying Data from Related Objects Using Nested Data Repeaters
- Dynamically Retrieving Data with Data Repeaters
- Dynamically Retrieving Data with Data Tables

Displaying Data from Related Objects Using Nested Data Repeaters

You can retrieve data from any child object of a parent object using a data repeater that contains another data repeater, data table, or data function. The outer or parent data repeater connects to an object, such as Accounts. In turn, the inner data repeater, data table, or data function automatically lets you connect to any child objects, such as Contacts. This is also known as a *nested data repeater*.

For example, if a data repeater is connected to Accounts, you can add a nested data function to it that's connected to Contacts to return the number of contacts associated with each account.

To create a nested data repeater:

1. Add a data repeater to the page.
2. Drag another **Data Repeater**, **Data Table**, or **Data Function** from the Page Elements pane onto the data repeater.
3. Select the related Salesforce object that you want to connect to.

   Note: You can also retrieve data from unrelated objects. However, as this can adversely affect the performance of your site, we recommend retrieving data from related objects only.

4. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
   a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *`parent_object_name.field_name`*.
   b. Select the operator to control how results are filtered. For example, select *`Equals`* to return an exact match.
   c. Select the source of the filter value. For example, to specify an explicit value, select *`Fixed value`*, or to use the values passed to the page via a query string, select *`URL query string`*.
   d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
   e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

   Note: If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

5. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
6. In Limits, you can limit the number of returned results. For example, if you're only interested in the top five results, enter *`5`* in the `Limit results to` field.
7. If you're adding pagination, specify the number of results to display per page in the `Results per page` field.
8. If you're working with a data table, click **Next** and add fields to the table by double-clicking a field, or selecting it and clicking >.
9. Click **Save**.

You can either add data elements or custom code to the parent repeater to display its fields. Similarly, if you've nested a data repeater inside the parent data repeater, add data elements or custom code to the nested data repeater to display the child object's fields.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

  AND

  "Manage Profiles and Permission Sets"

  AND

  "Customize Application"

Note: You can't nest data repeaters more than one level deep.

SEE ALSO:

Accessing Data in Related Objects Overview
Dynamically Retrieving Data with Data Tables
Using Data Functions

Improving Performance Using Caching

When working with data-bound page elements, such as data repeaters, data tables, and data functions, you can improve the performance and page rendering of your website using caching. Caching controls how often a page containing a data connection requests data from Salesforce.

Lets say 100 people visit the site page at the same time. Without caching, the page makes 100 separate requests for the same data, slowing performance considerably. However, with caching enabled, the data is requested and retrieved only once—the first time someone visits the page. Any subsequent requests for data during a set time period are returned from the cache. When the specified time period expires, the cache is refreshed.

The `Cache Duration (Minutes)` field in the Properties tab controls the length of time to cache retrieved data for the selected data repeater, data table, or data function.

The default value is 30 minutes. However, the appropriate number of minutes depends on your requirements. For example:

- If the data is updated frequently, such as in a commenting system or a stock ticker, you can disable caching by setting the value to zero to ensure the freshest data displays on the page.
- If the data changes infrequently, say just once a week, you can set the value to a much greater number of minutes. A longer caching period also helps ensure pages can display data even if the data source is momentarily unavailable.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Note: Whenever updates to a site are published, the cache is deleted for all data connections. Caching begins again the next time a site visitor accesses the page.

SEE ALSO:

Dynamically Retrieving Data with Data Repeaters
Dynamically Retrieving Data with Data Tables
Using Data Functions

Data Filters

When you add a data repeater, a data table, or a data function to a page, you don't have to limit the records it retrieves. However, if you're working with a Salesforce object that has thousands of records, you can limit the returned results using filter criteria.

When you add filter criteria, you need to specify:

- The field to which the filter criteria apply
- The operator
- The source of the filter value
- The filter value

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Filter Operators

Operator	Description
Equals	Returns an exact match.
Not equal to	Returns records that don't have the value you specify.
Less than	Returns records that are less than the value you specify.
Greater than	Returns records that exceed the value you specify.
Less than or equal to	Returns records that match or are less than the value you specify.
Greater than or equal to	Returns records that match or exceed the value you specify.
Starts with	Use when you know what your value starts with, but not the exact text. For example, "california" would return California Travel, but not Surf California.
Ends with	Use when you know what your value ends with, but not the exact text.
Contains	Returns records that include your search string but might also include other information. For example, "california" would return California Travel and Surf California.
Includes	Use for field types that support multiple values to return records that contain one or more of the comma-separated values you specify.
Includes all	Use for field types that support multiple values to return records that contain all of the comma-separated values you specify.
Excludes	Use for field types that support multiple values to return records that don't contain one or more of the comma-separated values you specify. For example, for a Locations category field, "San Francisco,Vancouver" would exclude a record containing "San Francisco,Dallas" and a record containing "Vancouver,New York."
Excludes all	Use for a field types that support multiple values to return records that don't contain all of the comma-separated values you specify. For example, for a Locations category field, "San Francisco,Vancouver" would exclude only records that contain *both* terms.

Filter Value Sources

Source	Description
Fixed value	Use when you want to specify the value.
URL query string	Use when you want to pass variable content via a URL to the item when the page loads.
Global property	Use when you want to use a fixed value from the site, such as the current date or current time.
Request header	Use when you want to use a value from the browser, such as the host header or browser version.

Source	Description
Parent repeater	Use when you want to create a query between unrelated objects. Available only when a data repeater, data table, or data function is nested inside a parent repeater, but the parent repeater's object is unrelated to the nested item's object. Warning: Retrieving data from unrelated objects can adversely affect the performance of your site. We recommend retrieving data from related objects only.

SEE ALSO:

- Dynamically Retrieving Data with Data Repeaters
- Dynamically Retrieving Data with Data Tables
- Using Data Functions
- About Displaying Dynamic Data Using Expressions

About Displaying Dynamic Data Using Expressions

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Site.com uses expression language to display data dynamically. Expressions serve as placeholders for data that will be replaced with information when the page loads. When working with data-bound page elements or custom widget properties, you can use expressions to customize how data is displayed on the page.

In Site.com, expression syntax consists of an open curly brace and exclamation point, the field, custom property name, or namespace name, and a closing curly brace.

For example, if you added a custom property called `URL` to a widget, you can use the syntax `{!URL}` to add the expression to custom code or content blocks. Similarly, to add an expression for the Billing City field, you can use the syntax `{!BillingCity}`. For related objects, the field name is prefixed by the name of the master object: `{!Account.BillingCity}`.

If you're editing a data element in a data repeater or a data table column, you can access the object's fields by name in a drop-down list without using expressions. However, if you want to customize how the field is displayed on the page, you can see and edit the expression when you click **Customize** in the dialog box.

If you're working with custom code or content blocks in a data repeater, you can only access the object's fields using expressions.

Expressions let you customize the output by:

- Adding text around the expression. For example, lets say you're displaying the phone number of each of your business locations. You could enter the text *`Contact us at`* before the `{!Phone}` expression. When the data is displayed on the page, `{!Phone}` is replaced with the field's value for each record: *`Contact us at 100-200-3000`*.
- Formatting the output using HTML tags. For example, you could wrap `H1` tags around the expression to alter how the output is displayed on the page: `<H1>{!Phone}</H1>`.
- Creating a URL query string to pass variable information to a data repeater or data table on another page. The second page, in turn, uses the received variable to retrieve and display the relevant records. For example, you could create a hyperlink, such as `/product_details?productID={!id}`, where `{!id}` is replaced with the product ID for each record. When a particular

product link is clicked, the product ID is passed to the Product Details page, which uses the ID to retrieve the record's information and display it on the page.

SEE ALSO:

Dynamically Retrieving Data with Data Repeaters

Dynamically Retrieving Data with Data Tables

Using Data Functions

Data Filtering Examples

When working with data repeaters, data tables, and data functions, you can filter the data you retrieve in many ways. In this topic, we explore two options—fixed values and URL query strings—to illustrate some common filtering techniques.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Example 1: Using a Fixed Value to Filter Results

In our first example, we have a custom object called "News" that stores company news and events. Each news item has a `Status` picklist that can be set to either In Progress or Approved. We only want to display approved news items on our Company News page. In this case, we can use a fixed value to filter the data in the News object.

In this scenario you would:

1. Add a data repeater to the page and configure it as follows:
   a. Select the News custom object.
   b. In the Filters section, set the criteria to `Status Equals Fixed value`.
   c. Enter *`Approved`* in the Value text box.

   This tells the data repeater to return only records where the `Status` field contains the value `Approved`.
2. Add data elements to the data repeater to display the required fields, such as Title, Description, and Date, on the page.

Example 2: Using a URL Query String to Dynamically Filter and Display Results on Another Page

In our second example, we have a custom object called "Products" that stores product information. However, some products are only available in certain locations, so we'd like to let customers view the products in their nearest city. In this case, we could create a Locations site page that contains links to each available city.

We want each link to open the Products site page, but only display products based on the user's selection. We can do this using URL query strings, which allow us to pass variable content between HTML pages. The variable information in this case is the product location; the data repeater won't know which products to return until the user makes a selection.

For this scenario, you would:

1. Add a data repeater to the Products page and configure it as follows:
   a. Select the Products custom object.
   b. In the Filters section, set the criteria to *`City Equals URL query string`*.
   c. In the Value text box, enter the variable name—in this case, *`location`*.
   d. In the If query string is missing drop-down list, select **Don't apply this filter item**. This option is used when a customer wants to view all products without filtering.
   e. Click **Save**.

2. Add data elements for the fields you want to display, such as Product Name, Description, and Price.
3. On the Locations page, add a data repeater and select the Products custom object.
4. Add a data element to the data repeater to represent the `City` field and configure it as follows:
   a. Select the `City` field as the field to display, because we want to use the name of the city as the hyperlink.
   b. Select Text as the display format.
   c. Select **Add a hyperlink** to display a URL on the page.
   d. In the Link to drop-down list, select An item in your site.
   e. Select Page as the type and select the Products page. (If you can't see the list of pages, place your cursor in the URL text box and press the DOWN key on your keyboard.)
   f. Change the URL value to `/product?location={!City}`. In this case, `{!City}` is a placeholder for the value of a record's `City` field. When the page first loads, `{!City}` is replaced with the correct value, such as Dallas, which creates the following URL for that record:

      ```
      /product?location=Dallas
      ```

      When clicked, this opens the Products page and passes `Dallas` as the value of the `location` variable.
   g. For the tooltip, select the `City` field and click **Customize**.
   h. Change the value to `Show me products available in {!City}`.

      Again, the `{!City}` placeholder is replaced by the value of the `City` field for each record when the page loads.

Now, when the Locations page loads, the data repeater displays the location of each product as a link. When a customer clicks a link, such as Dallas, the Locations page passes `location=Dallas` to the Products page. As the Products page loads, the data repeater uses this value to dynamically return only records where the `City` field contains the value `Dallas`.

SEE ALSO:

- Dynamically Retrieving Data with Data Repeaters
- Displaying Data Using Data Elements
- Dynamically Retrieving Data with Data Tables
- Accessing Data in Related Objects Overview

Collecting and Submitting Data Using Forms

Adding a Form to the Page

Use forms to collect data from your site visitors and submit the data to standard or custom Salesforce objects. Create web-to-lead forms, capture customer details, or gather feedback on your products or services.

To add a form to a page:

1. Drag a **Form** from the Page Elements pane onto the page.
2. Select the Salesforce object that you want to submit data to.

   Note:

   - For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's "Create" permission, and refresh the list.
   - For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Add available fields to the form by double-clicking a field, or selecting it and clicking >.

   All required fields are automatically added to the list of selected fields. However, you can hide required fields after you add the form to the page.

4. Reorder the list of selected fields by clicking **Move Up** or **Move Down**.
5. Click **Save**.

Note: When adding forms to authenticated community pages in Site.com, set the current user for Salesforce objects that require the Owner ID field. Setting the current user (as opposed to the default guest user) lets you identify the authenticated user when the form is submitted. To set the current user for the Owner ID field, select the field in the form, click **Configure** under Field Properties in the Properties pane, select `Global Property` as the source, and select `Current userID` as the value.

After you add a form to the page, you can't change the object it's connected to. If you need to connect to a different object, you must replace the form.

You can use the form's Properties pane to:

- See which object the form is connected to.
- Add a title to the top of the form.
- Specify what occurs when a user successfully submits the form.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add a form to the page:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned in Site.com Studio

To edit the guest user profile:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned in Site.com Studio

  AND

  "Manage Users"

  AND

  "Customize Application"

- Change the appearance of the form by selecting a different theme.

SEE ALSO:

Adding Input Fields to Forms or Pages

Input Field Types

Editing Input Fields in a Form

Adding Input Fields to Forms or Pages

Adding Input Fields to a Form

You can add additional input fields to an existing form. Each input field binds to a field in the object the form is connected to.

The quickest option is to:

1. Select the form on the page.
2. Select > **Add Fields**.
3. In the Add Fields list, click the fields that you want to add. The Add Fields list displays the available fields in the object that the form is connected to. When you click a field, the correct field type is automatically added to the form, such as a checkbox or picklist field.

Alternatively:

1. Select the form on the page.
2. Select > **Add Page Elements**.
3. In the Add Page Elements list, click the input field type that you want to add, such as **Checkbox**.
4. Choose a field in the Add a Field dialog box and click **Save**. If no fields of that type exist in the object, you can't add a field of that type to the form.

Note:

- You can't add fields to a form by clicking > **Edit Form**.
- Formula, encrypted text, geolocation, and lookup field types aren't supported.
- You can't configure the default field-level error messages that appear when users enter an incorrect value.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Input Fields to a Page

You can add input fields directly to a page, panel, data repeater, or data table to build your own custom features using custom code. For example, let's say some of your products are only available in certain locations and you'd like to let customers view the products in their nearest city. You could add a picklist input field to the page that lists the various locations. Using custom code, you could then pass the user's selection to a data table or data repeater via a query string to display a filtered product list.

To add an input field to the page, drag it from the Page Elements pane onto the page. Alternatively, select the page or container page element in the Page Structure pane, select > **Add Page Elements**, and select the input field.

SEE ALSO:

- Input Field Types
- Input Field Properties
- Editing Input Fields in a Form
- Adding a Form to the Page

Editing Input Fields in a Form

After you've added a form to the page, you can edit and reorder its fields.

To reorder fields, drag them to the correct position on the page or in the Page Structure pane. Alternatively, select a field and click > **Move Up** or > **Move Down**.

To hide a field, such as required field that you don't want your site visitors to see, select `Hidden Field` in the Properties pane. When you hide a field, you can access it from the Page Structure pane.

To make a field a required field, which means the user must complete the field before submitting the form, select `Required Field` in the Properties pane. A red asterisk (*) is displayed beside the field to indicate that it's required. You can't change the `Required Field` setting for any fields that are required by the object the form is connected to.

To rename a field, replace the name in the `Label Name` field in the Properties pane.

To change the appearance of a field on the page, select a different theme in the Properties pane. If the field is in a form, you can only change the form's theme.

Note:

- You can't move fields from a form onto the page. However, you can delete non-required fields, or hide both required and non-required fields.
- You can't drag fields from the page onto a form.

SEE ALSO:

- Adding Input Fields to Forms or Pages
- Input Field Properties
- Setting the Default Value of Input Fields
- Adding a Form to the Page

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Input Field Types

When adding input fields to a form or page, the following field types are available. However, if the object connected to a form doesn't contain a particular field type, you can't add that input field type to the form.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Page Element	Description
Checkbox	Lets users set a true (selected) or false (deselected) value on a form or page.
Currency	Lets users enter a currency amount on a form or page.
Date	Lets users enter a date on a form or page. If the field's `Date/Time Selector` property is set to Popup Calendar, the user can pick the date from a popup calendar.
Date/Time	Lets users enter a date and time on a form or page. If the field's `Date/Time Selector` property is set to Popup Calendar, the user can pick the values from a popup calendar.
Email	Lets users enter a valid email address on a form or page.
Multi-Select Picklist	Lets users select one or more values from a list on a form or page.
Number	Lets users enter a whole number on a form or page.
Percent	Lets users enter a percent amount on a form or page.
Phone	Lets users enter a phone number on a form or page.
Picklist	Lets users select a value from a list on a form or page.
Rich Text Area	Lets users enter up to 32,768 characters on a form or page. Supports any combination of letters, numbers, or symbols. Users can format the text, and add images and hyperlinks.  Note: A rich text area field can't be used for image uploads in Site.com sites or Force.com Sites due to security constraints.
Text	Lets users enter up to 255 characters (depending on the field limit in the Salesforce object) on a form or page. Supports any combination of letters, numbers, or symbols.
Text Area	Lets users enter up to 32,768 characters on a form or page, which display on separate lines. Supports any combination of letters, numbers, or symbols.

Page Element	Description
URL	Lets users enter a valid website address on a form or page.

SEE ALSO:

- Adding Input Fields to Forms or Pages
- Input Field Properties
- Adding a Form to the Page

Input Field Properties

When adding or editing input fields on a form or page, use the options in the Field Properties section of the Properties pane to control how the selected input field functions.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Name	Description
`Label Name`	The external name of the field. This is displayed as the field name on the form or page.
`Default Value`	Sets the field's default value. This automatically populates the field with the value you specify when the page loads.
`Required Field`	When selected, makes the field mandatory, so the user can't submit a record without entering a value. A red asterisk (*) is displayed beside the field to indicate that it's required. You can't change the `Required Field` setting for any fields that are required by the object the form is connected to.
`Hidden Field`	When selected, hides the field from the form or page. For example, you may want to hide a required field from your site visitors, or set the default value for a field that you don't want users to see or edit. When you hide a field on the page canvas, you can still access it from the Page Structure pane.
`Date/Time Selector`	Applies to the `Date/Time` field only. Sets whether users can select the date and time using a popup calendar.
`Picklist Values`	Applies to `Picklist` and `Multi-select Picklist` fields only, and only when added directly to the page. Sets the list of items to display in the picklist. (You can't modify the picklist items if the field is connected to an object.)

Name	Description
	Each picklist item consists of a `Label` and a `Value` field. The `Label` field is displayed in the picklist, whereas the `Value` field is an internal value. In most situations, you can use the same value in both fields, unless you want to submit a different value than the one displayed to the user. Click + to enter values for the `Label` and `Value` fields.
`Rows`	Applies to `Rich Text Area` and `Text Area` fields only. Sets the number of rows of text to display.

SEE ALSO:

- Input Field Types
- Adding Input Fields to Forms or Pages
- Editing Input Fields in a Form

Setting the Default Value of Input Fields

You can set the default value of an input field that you add to a form or page. This automatically populates the input field with the value you specify when the page loads.

When used with a field's `Hidden Field` property, which hides the field on the form or page, the default value is a useful tool for submitting data that you don't want your users to see. For example, you could add a hidden field that uses the Global Property option to track when a form is submitted.

Note: Default field values that are already set on the object aren't transferred to Site.com.

To set the default value of a field:

1. Select the field.
2. Click **Configure** in the Properties pane.
3. Select the source of the default value.

Option	Description
Fixed Value	Use when you want to specify the value. For example, for a text field, you could add default text. Alternatively, if the field is a picklist, you can select a default value from the list.
URL Query String	Use when you want to pass variable content via a URL to the item when the page loads. For example, if you're working with a picklist field, you could pass variable information, such as location, to filter the list and display only items from that location. See Data Filtering

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Option	Description
	Examples for similar scenarios using data repeaters.
Global Property	Use when you want to use a fixed value from the site, such as the current date or current time.
Request Header	Use when you want to use a value from the browser, such as the host header or browser version.

4. Set the default value.
5. Click **Save**.

SEE ALSO:

Adding Input Fields to Forms or Pages
Editing Input Fields in a Form
Adding a Form to the Page

Setting a Form's Submit Behavior

When your site visitors submit a form successfully, you can either redirect them to another page or display a message indicating that they were successful.

When the page is open:

1. Select the form in the Page Structure pane.
2. In the Properties pane, select an option in the When Successful drop-down list.

Option	Description
Show another page	Redirects the user to the page that you specify.
Display a message	Displays a message below the form.

3. As appropriate, specify either:
   - The page to redirect to in the `Page URL` field. You can enter a relative URL, such as a site page, or an absolute URL. For relative URLs, ensure you include a forward slash (/).
   - The message text in the `Message Text` field.

Tip: The form's **Submit** button uses the submit action in the Events pane.

SEE ALSO:

Adding a Form to the Page
Input Field Properties
Site.com Data Services Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Styling Forms

Forms are styled using CSS themes that you can customize to match the design of your website.

When you add a form to a page, the form uses a default theme to control its appearance. To change the form's theme, select an option in the Theme section in the Properties pane:

- `Default` places the field name above the text box. Required fields are denoted by a red asterisk (*) beside the field name.
- `Salesforce` places the field name to the left of the text box. Required fields are denoted by a red vertical bar (|) in front of the text box.

Alternatively, to customize a theme to suit your needs:

1. Select the form on the page.
2. Select a theme to use as a base in the Theme section of the Properties pane.
3. Open the Style pane and ensure `Class` is selected.
4. In the Style drop-down list, select the part of the form that you want to style. When you select an item, it's highlighted for a few seconds so you can easily see which part you're styling.

Option	Description
Entire form	Styles the whole form.
Field rows	Styles each field row.
Field labels	Styles the labels of each field.
Fields	Styles the fields (text boxes, drop-down lists, and so on). Only available for the Salesforce theme.
Required symbol	Styles the asterisk symbol (*) for required fields. Only available for the default theme.
Error message	Styles the error message that's displayed when users try to submit an incorrectly completed form. Only available for the default theme.

Tip: If you're familiar with CSS, you can also modify the style of the form in the site's style sheet.

5. To style the selected part of the form, update the Style pane properties. Your theme customizations are reflected immediately in the form, and apply to the selected form only.
6. Repeat as required for each part of the form.

SEE ALSO:

Adding a Form to the Page

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

The Default, Error, and No Data Views

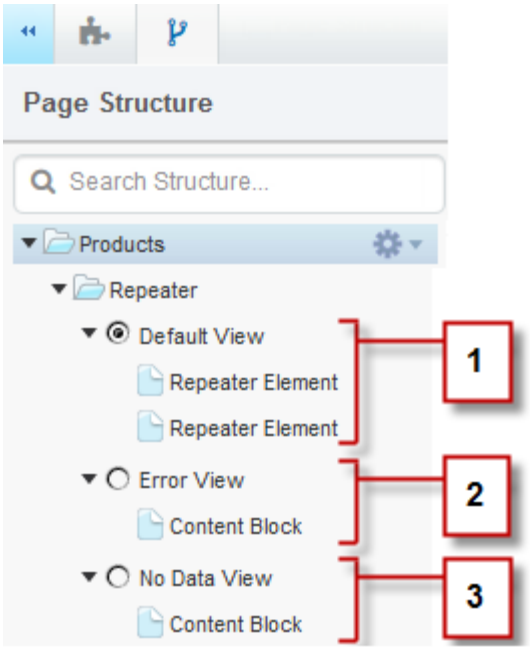
When working with data repeaters, data tables, data functions, and forms, you can customize what your site visitors see if an error occurs when connecting to the data source. Additionally, for data tables, data repeaters, data functions, you can customize what's displayed when no data exists for the current query. For example, if you set up a data repeater to dynamically filter and display results based on the user's selection, but there are no results for that selection, you can display an appropriate message to explain what happened.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition



- The Default View (1) is what usually appears on the page. For example, when you add a form to the page, the Default View contains the form's fields.
- The Error View (2) is displayed if an error occurs when connecting to the data source. It contains a content block with a default message that you can customize. You can also add other page elements to the view.
- The No Data View (3) is displayed when no data exists for the current query. It contains a content block with a default message that you can customize. You can also add other page elements to the view.

SEE ALSO:

Dynamically Retrieving Data with Data Repeaters

Dynamically Retrieving Data with Data Tables

Using Data Functions

Repairing Data Connections

If you or another user modifies the object that an existing data repeater, data table, data function, or form is connected to, the data connection might break. For example, this can happen if a connected object or field is renamed or deleted, or if its permissions are changed.

When you open a site page containing broken data connections, a dialog box that lists the problems appears. Hover over each item's icon to see the possible solutions.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Problem	Solution
A data repeater, data table, or data function is connected to an object that doesn't have the correct access permission, or has been renamed or deleted.	Open the guest user profile and ensure the "Read" permission is enabled on the object. If that's unsuccessful, click **Edit** beside the listed item to open the Data Connection dialog box, and ensure the correct object is selected. If the object has been renamed, select the renamed object. If the object has been deleted, you must either choose a different object or delete the page element.
A data repeater, data table, or data function is trying to filter results using fields that are longer visible, or have been renamed or deleted.	Open the guest user profile and ensure the object has the correct field-level security enabled for each field. If that's unsuccessful, click **Edit** beside the item to open the Data Connection dialog box, and reset the filter criteria.
A data element is connected to a field that's no longer visible, or has been deleted or renamed.	Open the guest user profile and ensure the object has the correct field-level security enabled for the field. If that's unsuccessful, click **Edit** beside the item to open the Edit Data Element dialog box, and ensure the correct fields are referenced in all relevant drop-down lists and in any custom text.
A form is connected to an object that doesn't have the correct access permission, or has been renamed or deleted.	Open the guest user profile and ensure the "Create" permission is enabled on the object. If that's unsuccessful, you must replace the form, as you can't edit a form's data connection.
A form is missing one or more required fields, which were added to the object after the form was created.	Add the missing field to the form.
A form field is pointing to a field that's no longer visible, or has been renamed or deleted.	Open the guest user profile and ensure the object has the correct field-level security enabled for the field. If that's unsuccessful, remove the field from the form.

Note: If you're a Communities user working with authenticated pages, keep in mind that object access on public and private pages is controlled by the user profile of the authenticated user. The guest user profile controls object access on public pages for unauthenticated visitors only.

SEE ALSO:

Setting Data Access Permissions for Salesforce Objects

Dynamically Retrieving Data with Data Repeaters

Adding Input Fields to Forms or Pages

Building Custom Widgets

Widgets Overview

Widgets let you save time by building custom page elements that you and your team can reuse throughout the site.

Using the existing Site.com page elements, such as panels, content blocks, custom code, or data repeaters, you can create widgets to suit your unique requirements. You can add custom properties to allow greater flexibility over how your widgets are reused. And you can even add a widget to another widget!

By using CSS to style the widget, you can ensure it always appears correctly whenever it's added to site pages or page templates. Additionally, you can let contributors add widgets to site pages and add branding properties that enable contributors to update the widget's appearance.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

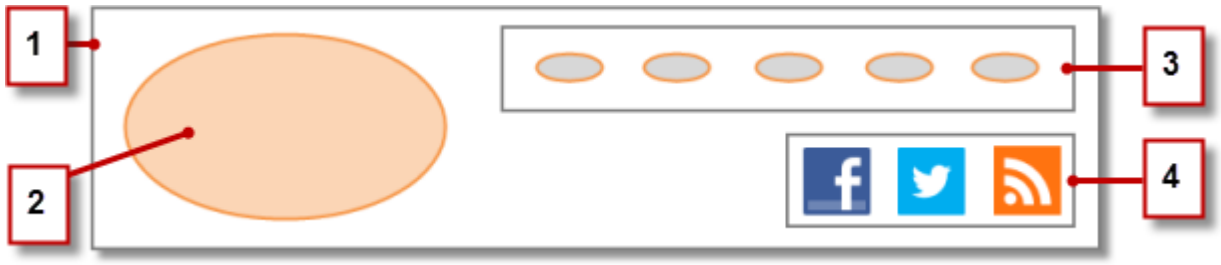
Available (with limitations) in: **Developer** Edition

With widgets, you can:

- Minimize duplication in your site. You just build once, then reuse.
- Reduce maintenance overheads for you and your team. Any updates you make to a widget are automatically reflected in copies of the widget on the page.
- Improve the load time of your pages.

Example: For example, you could use the custom code page element to create reusable social plug-in widgets, such as **Like** or **Follow** buttons from Facebook, or a Twitter feed. Like

Alternatively, to ensure a consistent look and feel across all of your pages, you could create a corporate header such as this sample header widget. It consists of a panel (1) that contains a company logo (2) and a menu (3). The widget also contains another widget (4) that's composed of a panel containing custom code for Facebook, Twitter, and RSS plug-ins.



You could also use widgets to store commonly used pieces of text, such as company names, addresses, legal text, and so on. Simply create a widget that contains a content block with the relevant text.

Best Practices for Using Widgets

- Wherever possible, use widgets to reduce duplication in your site design. This is particularly important when working with complex designs or site elements, where maintenance can be time consuming.
- If you plan to use the same widgets across several sites, consider creating a basic site to contain all the required widgets. Then, for each new site, just create a copy of that basic site. That way, each new site automatically includes all of the widgets you created.
- When using CSS to style widgets, add the CSS to the Site Style Sheet, which is the site's global style sheet. Because every page automatically references the style sheet, you can ensure that each widget will appear correctly on the page.

SEE ALSO:

Creating Widgets

Custom Properties for Page Templates or Widgets Overview

Site Branding Overview

Creating Widgets

Widgets let you create custom, reusable page elements by combining existing Site.com page elements, custom code, and CSS.

When you create a widget, it's added to the Widgets view on the Overview tab, where you can access and manage all of the site's widgets. If you make it available, the widget also appears in the Widgets section of the Page Elements tab, where you and your team can easily drag it onto the page. You can also let contributors add widgets to site pages.

1. Hover over **Widgets** on the Overview tab and click **New**, or click **New Widget** when the Widgets view is open.
2. Enter the widget name. This is name that appears in the Page Elements pane.
3. Optionally, add a description.

   The description appears as a tooltip in the Page Elements pane.
4. To add a display icon for the widget, select an image from your imported assets. Use an icon that's 16 x 16 pixels in size.
5. Optionally, deselect `Available in the Page Elements Pane` if you don't want the widget to appear in the Page Elements pane.

   For example, you may not want the widget to appear until you've finished building it.
6. Click **Apply**. Now you're ready to add page elements to the widget.
7. Click the widget's name to open it in a new tab.
8. Add the page elements and CSS styles you need.
9. Optionally, add custom properties or branding properties to the widget.

After you finish building the widget, ensure it's available in the Page Elements tab by selecting `Available in Page Elements Pane` on the Properties pane.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To let contributor's add a widget to a site page, select `Available to Contributors` on the Properties pane. This setting controls whether the widget appears in the contributor's Page Elements pane. You must also ensure that the site includes a template-based site page with at least one editable panel.

SEE ALSO:

- Widgets Overview
- Custom Properties for Page Templates or Widgets Overview
- Site Branding Overview
- Adding a Widget to a Page

Adding a Widget to a Page

Before you can add a widget to a page, you must make sure it's available for use. Either:

- Select `Available in Page Elements pane` in the Properties pane when the widget is open.
- Hover over the widget in the Widgets view on the Overview tab, select > **Edit Properties**, and select `Available in Page Elements pane`.

When the page is open:

1. Drag the widget from the Widgets section of the Page Elements pane onto the page. When you add a widget to a page, it creates a copy or *instance* of the widget. You can't edit this widget instance.
2. If available, update the properties in the widget instance's Properties pane.

SEE ALSO:

- Widgets Overview
- Custom Properties for Page Templates or Widgets Overview
- Editing and Deleting Widgets
- Creating Widgets

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Editing and Deleting Widgets

Access the site's widgets in the Widgets view under All Site Content (on the Overview tab).

To update a widget, hover over it and click . From the menu, you can:

- Edit the widget. When you edit a widget, it opens in a new tab. Any updates you make are reflected immediately in the site pages or page templates that reference it.
- Update the widget's properties, including the name, description, display icon, and availability. In the Widgets view, you can also edit the name, description, or availability inline by double-clicking the item.
- Duplicate, preview, or delete the widget. You can't delete a widget if it's being used on a site page, page template, or another widget.

SEE ALSO:

Widgets Overview

Creating Widgets

Adding a Widget to a Page

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

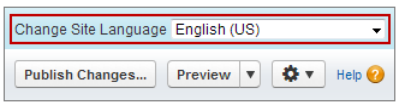
  Site administrator or designer role assigned at the site level

Creating Multilingual Sites

Multilingual Sites Overview

Site.com Studio lets you to create different language versions of your site. And because all languages are maintained within the site, you don't need to create and manage a separate site for each language.

As a site administrator or designer, you can add and manage the languages that you want your site to support. After you add a language to the site, you'll notice the Site.com Studio language selector on the toolbar, which lets you and your team switch between languages when editing the content of a page. This makes content editing quick and easy because you never have to leave the page to change to another language.



EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Contributors, designers, and site administrators can each add language-specific content to a page using the language selector.

As a site administrator or designer, you can also export site content as an .xml file and send it to your translation service. After you receive the edited file, simply import it back into your site to populate each page with the translated content.

In turn, to let your site visitors choose their preferred language from those available when viewing the site, you can add a Language Selector page element to your pages. And in case any part of the site content isn't available in their chosen language, you can specify a *fallback* language to display instead.

Example: For example, if a site visitor chooses French (CA) from the language selector, but there is no content for that page in French (CA), content in the fallback language—say, French (FR)—is displayed instead.

SEE ALSO:

Create a Multilingual Site

About Editing Language Content

Create a Multilingual Site

Creating a multilingual site is a multistep process. It involves defining the languages you want your site to support, adding translated content for each language, and letting your site visitors choose their preferred language.

1. Set the default site language. It's important that you do this before you begin adding translated content to your site.
2. Add languages to the site.
3. Set options for each language, such as the display language and fallback language.
4. Add content for each site language one of two ways:
   - Edit the page content for each language directly.
   - Export the content for translation and then import the translated content.
5. Add a Language Selector page element to your site pages, so authenticated site users can choose their preferred language.
6. If you have a self-service community built on the Customer Service (Napili) template, add the Language Picker component to your community pages. This allows guest users (users who aren't logged in) to choose their preferred language. For details, see Using Templates to Build Communities.

Tip: After you add language-specific content to your site, you can share a separate preview URL for each language. Just switch to the desired language in the Site.com Studio language selector and click **View Anonymous Preview**. Then copy the link to send it to your reviewers.

SEE ALSO:

Multilingual Sites Overview

About Editing Language Content

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit language content:

- `Site.com Publisher User` or `Contributor User` field enabled on the user detail page and any role assigned at the site level

Setting the Default Language

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages.

It's important to set the default language before you add any language content to your site. This setting is not associated with the default language setting in your Salesforce organization.

1. Click **Site Configuration** > **Languages** on the Overview tab.
2. Select a language in the Default Site Language drop-down list.

Note: If you decide to change the default language after you add translated content, you must first export the translated content, then change the default language, and finally, import the content back into your site. Otherwise the translated content won't appear for the newly selected default language.

For example, let's say you make English the default site language and add French as a site language. After you add content in both English and French, you decide to change the default site language to French. To preserve the French content, you must first export it. Then, select French as the default site language before importing the French content back into the site.

SEE ALSO:

Adding Languages
Setting Language Options

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding Languages

Add the languages you want your site to support.

1. Click **Site Configuration** > **Languages** on the Overview tab.
2. Click **Add Languages**.
3. Select the languages you want to add to your site.
4. If necessary, reorder the list as you want it to appear in any language selector.
5. Save your changes.

SEE ALSO:

- Setting the Default Language
- Setting Language Options
- Deleting Languages

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Setting Language Options

After you add site languages in the Languages view, you can define separate settings for each one.

1. Click **Site Configuration** > **Languages** on the Overview tab.
2. Select a language in Site Languages.
3. Update the options in Language Settings:

Option	Description
Active on Live Site	If you add a language selector element on your active site, this checkbox controls whether the language shows in the drop-down list. Use the property to hide a language until you're ready to release the associated content to your site visitors.
Fallback Language	The fallback language displays when there's no content available for the currently selected language. For example, if a site visitor chooses Japanese from the language selector, but there is no content for that page in Japanese, content in the fallback language is displayed instead.
Display Label	You can define the display label for each language. It appears in any language selectors you add to your site, and in the Site.com Studio language selector.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

SEE ALSO:

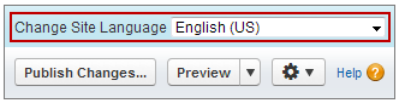
Adding Languages

Setting the Default Language

About Editing Language Content

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages. So for example, if you build your site and then add French as a site language, the French version initially contains English (US) content until you replace it with French-specific content.

After you add a language to your site, the Site.com Studio language selector appears on the toolbar. It lets you switch between languages as you edit content on each page. For example, if you add French and Spanish to your site, the language selector displays French, Spanish, and English (US).



EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Content blocks display icons to let you know whether content for the selected language has been replaced.

- A warning icon () indicates that the text that has not yet been replaced.
- A globe icon () indicates that the content has been replaced.

In addition, several page elements have *language aware* properties, meaning they can store different values for each language. These properties are indicated in the Properties pane by a globe icon ().

You can add content for each site language one of two ways:

- Edit the page content for each language directly using the Site.com Studio language selector.
- Export the content for translation and then import the translated content back into the site.

Language Display Order

After you add language-specific content to your site, content on the page is displayed in the following order, depending on which language content is available:

1. Selected language
2. Fallback language
3. Default language

For example, if you delete the French contents of a content block, and Spanish is the fallback language, the Spanish content is displayed rather than the default language content. In turn, if you delete the Spanish content, then the default language content is displayed.

Language–Aware Page Elements

The following page elements support translated content:

- Content Block
- Custom Code
- Data Element
- Form
- Input Fields
- Image
- Language Selector

SEE ALSO:

Language-Aware Properties

Editing Language Content on the Page

The Site.com Studio language selector lets you switch between languages as you edit content on each page. Contributors, designers, and site administrators can each add language-specific content to a page using the language selector.

When the page is open:

1. Select a language in the Site.com Studio language selector (the Change Site Language drop-down list on the toolbar).
2. Edit the page content.
   If you're viewing a content block in a specific language:
   - A warning icon ( ) indicates that the text that has not yet been replaced.
   - A globe icon ( ) indicates that the content has been replaced.
3. Optionally, update the language-aware properties as required.

Note: Site.com Studio does not validate languages as you enter content. Take care to add the correct content for the selected language.

To revert translated content, right-click the content block and select **Revert Translation**. Text reverts to the fallback language, if specified, or else to the default site language.

SEE ALSO:

About Editing Language Content

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit language content:

- `Site.com Publisher User` or `Contributor User` field enabled on the user detail page and any role assigned at the site level

Language-Aware Properties

Several properties in the Properties pane are *language aware*, meaning they can store different values for each language.

If you don't specify language-specific properties for a page or page element, the property values of the default languages are used instead.

Property	Applies To	Description
`Alternative Text`	Image	Used by screen reader users or as a substitute if the browser can't display the image. It can also help with search engine optimization (SEO).
`Do Not Translate`	• Content Block • Custom Code	Prevents the item's content from being translated. When selected, the globe icon ( ) is removed from language-aware properties, indicating that they no longer support language-specific content.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Property	Applies To	Description
	• Data Element • Form • Input Fields • Image • Language Selector • Page	If you add language-specific content and subsequently enable this property, your language-specific content is simply hidden. If you disable the property again, the content reappears.
`Home Page URL`	Language Selector	Redirects site visitors to a different page when they choose a language. The URL is the same for all languages.
`Image Asset`	Image	Lets you specify the image to display for the selected language. For example, with English (US) selected in the Site.com Studio language selector, select the English version of an imported image. Then, to enter a French version of the same image, select French in the language selector and select the French image.
`Label`	Language Selector	The label that appears beside the Language Selector page element when it's added to a page. "Change Site Language" is the default text. The text is translatable, so you can either choose a language in the Site.com Studio language selector and update the text for each language, or export all site content for translation.
`Navigation Name`	Page	The page name that appears in a navigation menu.
`Title`	Page	The title that appears in the title bar of browser windows.
`Visible in Live Site`	Page	If you add a menu to your site, controls whether the page appears in the menu for the selected language. Additionally, the page is no longer accessible via its language-specific URL. If `Do Not Translate` is also enabled, the status of `Visible in Live Site` applies to all languages.

SEE ALSO:

Editing Language Content on the Page

About Editing Language Content

Exporting Language Content

Export one or more site languages as an .xml file that you can send to your translation service.

1. In the Languages view on the Overview tab, click ▾ > **Export Content for Translation**.
2. Select the language content that you want to export for translation. (Use CTRL+click to select multiple items.)
3. Optionally, enter a different filename. The default name is `languages.xml`.
4. Click **OK**.
5. If required by your browser, choose where to save the file.

After you export the .xml file, you can send it for translation.

Example: The .xml file contains a time stamp attribute that records the time of the export, and encloses all translatable content in CDATA sections, as shown in this example.

```
<?xml version="1.0" encoding="utf-8"?>
<languagedata exporttimestamp="1375387451589" version="1.0">
  <component id="c5f290a5-0fb5-4679-95b7-ab27c940a9b5" context="Content Block (Content Block)">
    <field name="HTML" language="en_US" exportcrc="1113705739"><![CDATA[
    <p>This is our English content.</p>
  ]]>   </field>
    <field name="HTML" language="fr_FR" exportcrc="2453234912"><![CDATA[
    <p>This is our French content.</p>
  ]]>   </field>
  </component>
  <component id="8ef488bc-9040-49e9-9aa7-88b287cf5345" context="ffff (sites.data.vocabulary.Term)">
    <field name="TermDescription" language="en_US" exportcrc="0"><![CDATA[]]>   </field>
  </component>
</languagedata>
```

When you receive the translated .xml file, import it back into the site.

SEE ALSO:

- Importing Translated Content
- About Editing Language Content

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Importing Translated Content

You can import a translated .xml file back into your site after your translation service has completed the translations.

1. In the Languages view on the Overview tab, click ▾ > **Import Translated Content**.
2. Browse to the location of the translated .xml file.
3. Select the file and click **Open**.
4. Decide whether to overwrite the current site content.
5. Click **Import**.
   A message appears to indicate whether the content was imported successfully, unless the file is over 1 MB. In that case, you'll receive an email when the import process finishes.

After you import the translated content, test your pages to make sure the content displays correctly. For each page, use the Site.com Studio language selector to view the page in each supported language.

SEE ALSO:

Exporting Language Content
About Editing Language Content

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Adding a Language Selector Page Element

The language selector lets your site visitors choose their preferred language when viewing a site.

After you add languages to your site, you need to let site visitors select their preferred language from the list of languages you defined in the Languages view.

Tip: To save time, add the language selector to your site's page templates.

When the page is open:

1. Drag a **Language Selector** from the Page Elements pane onto the page.
   Site visitors will see a Change Site Language drop-down list when they visit the page.
2. Set properties for the language selector:

Property	Description
`Do Not Translate`	Select this checkbox if you want the text in the Label field to remain the same for all languages.
`Label`	The label that appears beside the Language Selector page element when it's added to a page. "Change Site Language" is the default text. The text is translatable, so you can either choose a language in the Site.com Studio language selector and update the text for each language, or export all site content for translation.
`Home Page URL`	Redirects site visitors to a different page when they choose a language. The URL is the same for all languages.

SEE ALSO:

Language-Aware Properties

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Deleting Languages

When you delete a language, the translated content is not actually deleted—it's just no longer available to your or your team. As soon as you add the language back to the site, you can access the translated content again.

1. Click **Site Configuration** > **Languages** of the Overview tab.
2. Click  beside the language you want to delete.

SEE ALSO:

Adding Languages

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and manage language options:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Using Events to Create Animated Effects

Events Overview

Events enable you to add interactive and animated effects to the pages and page elements of your website.

When an event occurs—say, when a user clicks an element on the page or when the page loads—you can specify what action (or series of actions) it should trigger. For example, when your home page loads, you could display a news bulletin popup that fades away after several seconds. Or when the user clicks a panel, you could expand the panel to reveal additional information or alter its style.

The Events pane lists many common actions that you can trigger when an event occurs. Using the options here, you can specify that:

When *[this event]* occurs, trigger *[this action]*.

SEE ALSO:

Creating an Event

Available Events and Actions

Using Site.com Studio as a Site Administrator or Designer

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Creating an Event

Use events to add interactive and animated effects to the pages and page elements of your website.

1. Select the relevant page or page element.
2. Select an event in the Events pane ( ). See Available Events and Actions on page 796.
3. Click + and select an action in the Choose an Action list that appears.
4. Set the other available properties for the action, such as:
   - Target Element—specifies the page element that the action affects
   - Effect—specifies how the action is animated, such as fade or slide.
   - Speed—sets the speed of the animation to fast, normal, slow, or very slow
   - Chained—allows you to chain actions so they occur sequentially. For example, to create a Delay action that delays the action that *follows* it, select the Delay action's **Chained** checkbox. This indents the subsequent action underneath the Delay action, indicating that it's tied to the Delay action. Applies to the Animate, Delay, Hide Element, Repeat, Show Element, and Toggle Element actions.
5. Click **Save**.
6. Add more actions if required.

Tip:

- When you add an event to a page or page element, an asterisk (*) appears beside the event in the Events pane.
- If you hover over on a selected page element, a tooltip appears indicating which events are associated with the element. You can also click the icon to quickly open the Events tab.

To delete an action, select it and click .

To change the order in which an action occurs, select it and click or .

SEE ALSO:

Adding Pagination to Data Repeaters and Data Tables

Events Overview

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Available Events and Actions

Choose from several event triggers and actions when you create an event.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When This Event Occurs...

Event	Occurs When...
Double click	The user double-clicks the page element.
Click	The user clicks the page element.
Focus	The focus moves to the page element.
Load	The page or page element is loaded in a browser window.
Blur	The focus moves from the page element.
Mouse in	The user moves the mouse pointer over the page element.
Mouse out	The user moves the mouse pointer out of the page element.

Trigger This Action...

Action	Description
Add CSS Class	Dynamically adds a CSS class to style the targeted item. For example, to alter the appearance of a page element, you could add a new CSS class to it.
Alert	Displays a popup browser alert message.
Animate	Animates CSS properties, such as Top, Left, Width, and Height, which you specify by entering appropriate values in the `CSS` field. For example, if targeting an image, you can enter values such as `opacity: "0.4", width: "70%"`, which changes the image's appearance according to the speed and effect you set.
Delay	Adds a delay (measured in milliseconds) before the action that follows. (Ensure you select the **Chained** checkbox to tie it to the subsequent action.)
Execute JavaScript	Runs custom JavaScript code, which you enter by clicking **Edit Script** to open the Custom Code Editor.
Go To Page	Goes to the designated page number in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables on page 757.
Hide Element	Hides the targeted item according to the speed and effect you set.
Next Page	Goes to the next page in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables on page 757.
Previous Page	Goes to the previous page in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables on page 757.
Remove CSS Class	Removes a CSS class from the targeted item to dynamically remove its style. For example, to alter the appearance of a page element, you could remove the CSS class associated with it and replace it with another.

Action	Description
Repeat	Repeats the action that follows by the specified number of times, with the specified delay between each occurrence. (Ensure you select the **Chained** checkbox in the Properties pane to tie it to the following action.)
Set Element Attribute	Dynamically sets the specified attribute value of the targeted item. For example, if targeting an image, you could change the image source by entering `src` in the `Attribute Name` field and entering the image URL in the `Attribute Value` field. You can also add custom name/value pairs for advanced coding purposes.
Show Element	Reveals the targeted item according to the speed and effect you set.
Submit	Submits the selected form's data.
Toggle Element	Switches the visibility of the targeted element according to the speed and effect you set.

SEE ALSO:

Adding Pagination to Data Repeaters and Data Tables

Events Overview

Editing Sites as a Contributor

Understanding the Contributor's Page Editing View

Open a site page on the Overview tab by double-clicking the page or hovering over it and clicking > **Edit**. The page opens as a new tab.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

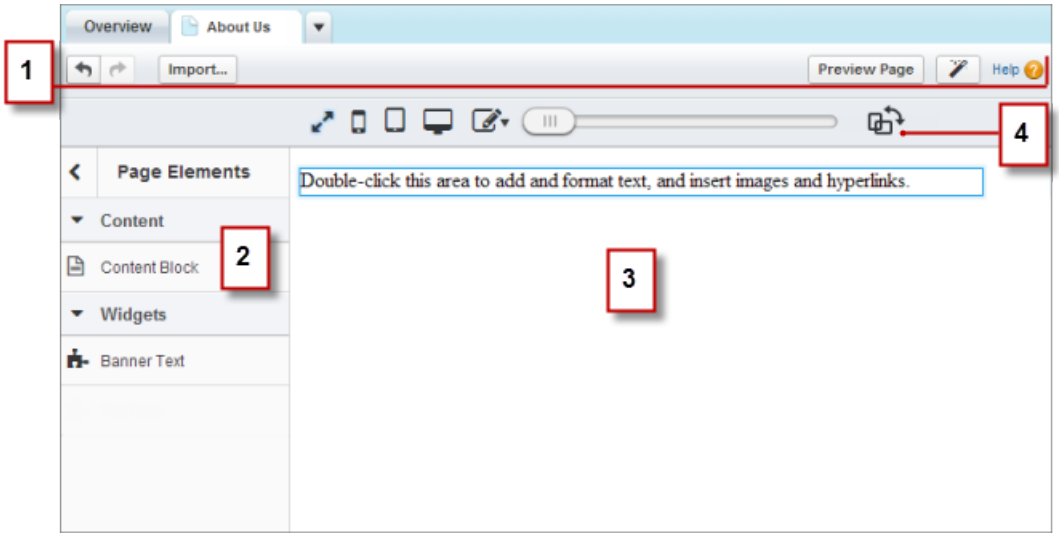
USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level



- Using the toolbar (1), you can:
  - Undo and redo your actions.
  - Import assets, such as images and files.
  - Preview the page in a browser window.
  - Update the appearance of the page using the Branding Editor.
- Using the Page Elements pane (2), you can drag content blocks and widgets (if available) to editable areas of the page.
- On the page canvas (3), you can edit page text and add images. If editable areas are available, you can drag page elements to the page.
- Using the live mode options (4), you can see how the page appears on various devices when the page is live.

SEE ALSO:

Creating Site Pages as a Site.com Contributor

Using Site.com Studio as a Contributor

Understanding the Inline Editing Toolbar

Creating Site Pages as a Site.com Contributor

If your site administrator or designer has enabled page creation, you can add pages to your site.

To add a new page:

1. In the Site Pages view on the Overview tab, click **New Site Page**.
2. Enter the page name.
3. Select a template for the page.
4. Click **Create**. The site page opens.

SEE ALSO:

Using Site.com Studio as a Contributor

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

Editing Content Blocks as a Contributor

Content blocks contain the text of your website pages, and can also house images and hyperlinks. As a contributor, you can edit the text in editable content blocks using an inline editor. Because you're editing the text inline, you always know exactly how the finished page will look when it's live.

Editable content blocks display a gray border when you hover over the text. You can edit the text in these areas only.

When the page is open:

1. Double-click the text area you want to edit.
   The inline editor appears.
2. Add or edit text, and format it using the inline editor.

   Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. This also ensures that all page text is updated automatically if a site administrator or designer modifies the site's paragraph and heading styles.
3. Add images or hyperlinks as required.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

Your changes are saved automatically when you click anywhere outside the text area.

SEE ALSO:

- Previewing How Pages Appear on Mobile Devices
- Understanding the Contributor's Page Editing View

Understanding the Inline Editing Toolbar

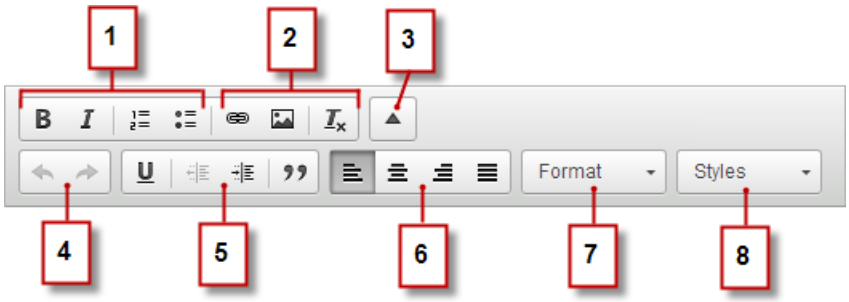
As a contributor, you can use the inline editor to edit any editable text areas on the page. Editable areas display a gray border when you hover over the text.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition



The inline editor lets you:

- Control the text style and layout (1) by:
  - Applying bold or italic formatting
  - Inserting numbered or bulleted lists
- Add images and hyperlinks, and remove unwanted formatting (2).
- Expand the toolbar (3) to access additional options (4–8).
- Undo and redo your edits (4).
- Control the text style and layout (5) by:
  - Applying underline formatting
  - Setting text alignment
  - Applying blockquote formatting
- Set paragraph indentation (6).
- Apply paragraph and heading styles (7).
- Apply additional styles, such as highlighting (8).

Tip: Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. This also ensures that all page text is updated automatically if a site administrator or designer modifies the site's paragraph and heading styles.

SEE ALSO:

- Editing Content Blocks as a Contributor
- Previewing How Pages Appear on Mobile Devices
- Understanding the Contributor's Page Editing View

Adding Images to Text as a Contributor

As a contributor, you can quickly add images to any editable text areas on the page using an inline editor.

Editable text areas display a grey border when you hover over the text. You can add images to these areas only.

When the page is open:

1. Double-click the text area you want to edit.
   The inline editor appears.
2. Position your cursor where you want to insert the image and click .
3. In the Image Properties dialog box, enter a URL to an image in the `URL` field.
   For example, to add an image that you uploaded to the site, type **/** followed by the image filename, such as *`/myimage.png`*.
4. Enter a brief description of the image in the `Alternative text` field. The description is used by screen reader users or as a substitute if the browser can't display the image. It can also help with search engine optimization (SEO).
5. Optionally, preview how the image appears in relation to the text on the page and set:
   - The width and height of the image
   - The image border (for example, to set a border that's 10 pixels wide, enter *`10`* in the `Border` field)
   - How much space surrounds the image (which is controlled by the `HSpace` and `VSpace` properties)
   - How the image aligns with the text on the page
6. Click **OK**.

Your changes are saved automatically when you click anywhere outside the text area.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

SEE ALSO:

- Editing Content Blocks as a Contributor
- Understanding the Inline Editing Toolbar
- Previewing How Pages Appear on Mobile Devices
- Understanding the Contributor's Page Editing View

Attaching Hyperlinks to Text and Images as a Contributor

As a contributor, you can use the inline editor to quickly add hyperlinks to text or images in any editable text areas on the page.

Editable text areas display a gray border when you hover over the text. You can add hyperlinks in these areas only.

When the page is open:

1. Double-click the text area you want to edit. Editable text areas display a gray border when you hover over the text.
   The inline editor appears.
2. Select the text or image that you want to attach a hyperlink to and click .
3. Select the link type.
   - To link to a page or item in your site, or to an external page:
     a. Select **URL**.
     b. Select the protocol.
     c. In the `URL` field, select an item in your site, or type the address to an external page—for example, `http://www.externalsite.com`. (If you can't see the list of site items, place your cursor in the `URL` field and press the DOWN key on your keyboard.)
   - To link to an anchor that was previously added to the page, select **Link to anchor in the text** and select the anchor in the drop-down list.
   - To link to an email message, select **An email**, and enter the recipient's email address and the message information.
4. Click **OK**.

Your changes are saved automatically when you click anywhere outside the text area.

SEE ALSO:

- Editing Content Blocks as a Contributor
- Understanding the Inline Editing Toolbar
- Previewing How Pages Appear on Mobile Devices
- Understanding the Contributor's Page Editing View

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

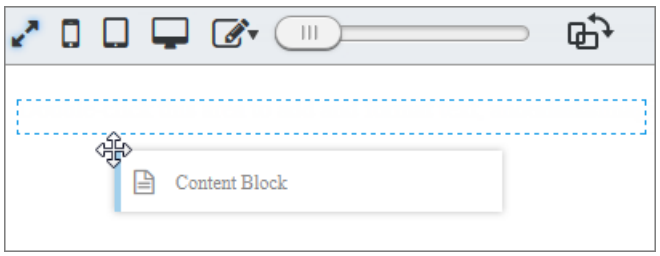
  Contributor role assigned at the site level

Adding Page Elements to Pages as a Contributor

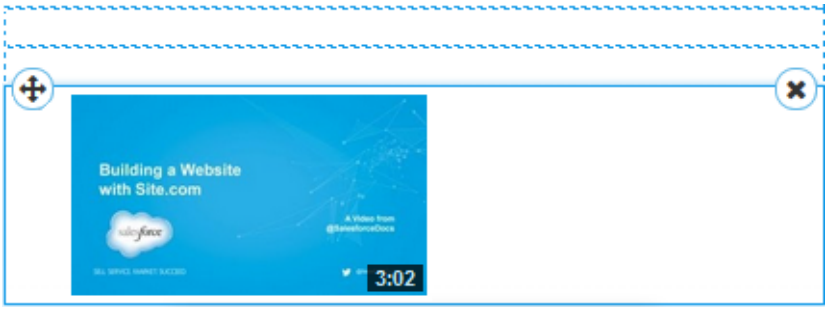
As a contributor, you can add page elements to any editable areas of a page.

Page elements are the building blocks of site pages. Content blocks contain the text of your website pages and can house images and hyperlinks. Widgets are custom page elements that are created by your designer or site administrator. If you can't see widgets, they're not available to you.

- To add a page element to an open page, drag the element from the Page Elements pane to an editable area. When you drag a page element to an editable page area, the area is highlighted with a blue border. If you can't see a blue border, the page doesn't have any editable areas.



- To move a page element on the page, drag the page element to another editable area.



- To delete a page element, click  .

After you add a content block to a page, double-click the content block to edit it.

When you add a widget to the page, you might need to specify properties for the widget in a dialog box, which automatically saves your changes. To dismiss the dialog box, click another area of the screen. For more information about the properties, contact your designer or site administrator.

SEE ALSO:

Editing Content Blocks as a Contributor

Understanding the Contributor's Page Editing View

Previewing How Pages Appear on Mobile Devices

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To edit only content in Site.com sites:

- `Site.com Contributor User` field enabled on the user detail page

  AND

  Contributor role assigned at the site level

Previewing How Pages Appear on Mobile Devices

With live mode, site administrators, designers, and contributors can preview how pages and templates appear on devices such as cell phones and tablets.

Site.com contributors are automatically placed in live mode for all content editing. They can edit the content blocks they've been given access to by the site administrators.

Site administrators and designers can switch between live and design modes by clicking **Live** or **Design** on the Site.com Studio toolbar. Design mode is the default view for site administrators and designers.

- To preview how the page looks:
  - On smartphones, click  .
  - On tablets, click  .
  - On computer screens, click  .
  - At 100% width and height, click  .
- To manually adjust the width of the screen, move the slider  .
- To resize the width and height of the screen, drag the resizing handles at the edge of the frame to the required size.
- To store frequently used custom screen sizes, click  and enter the custom measurements.
- To rotate the screen orientation from portrait to landscape, click  .

SEE ALSO:

Editing Content Blocks as a Contributor

Understanding the Contributor's Page Editing View

Editing Site.com Pages as a Designer or Site Administrator

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

Previewing Site.com Sites

Contributors, designers, and site administrators can preview site pages to see how they look when rendered in a browser window. It's always a good idea to make sure your changes are displayed correctly, as this is how the pages appear on the live site.

If you're a site administrator or designer, you can also create an anonymous preview URL that allows other users to review the site before it goes live. The URL is always valid (unless you disable it) and shows the latest work in progress. It's only available to the people you send it to, and can't be found by search engines.

- Click **Preview Page** on the toolbar when editing a page.
- Hover over a page in the Site Pages view of the Overview tab and click > **Preview** to view a single page.
- Click **Preview** on the toolbar of the Overview tab to view the entire site. From the **Preview** menu, you can also:
  - Click **Preview Site in a New Tab** to view the site in a new tab in the existing browser window.
  - Click **Preview Site at 1024 x 768** to view the site as it appears to laptop users.
  - Click **Enable Anonymous Preview**, if you're a site administrator or designer, to create a URL that allows other users to preview the site before it goes live. Click the **View Anonymous Preview** option that appears in the **Preview** menu to access the preview URL, which you can copy and send to other users to review and test your changes. **Enable Anonymous Preview** is also available in the Site Configuration view.
- Click **Preview** beside a site on the Site.com home page to view the entire site.

When you preview pages, all browser-related functions should work.

Note: During preview only, style sheets are rendered as inline styles.

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

Using Site.com Studio as a Contributor

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To build, edit, and manage Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

- `Site.com Contributor User`

  AND

  Contributor role assigned at the site level

Controlling Access Using IP Restrictions

Site.comIP Restrictions Overview

Every computer has a unique IP address that it uses to identify itself. Using IP restrictions, you can define a range of permitted IP addresses for the pages, folders, and assets in your site to control visitors' access.

For example, let's say you have a site page that lists all employees by department. You don't want people outside your organization to view this sensitive information. By restricting the permitted IP addresses to your organization's IP range—say, 112.122.0.1 to 112.122.0.123—you ensure that no other site visitors can view the page.

When you define IP restrictions for:

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

- A parent page template, all child page templates and template-based pages inherit the restrictions.
- A folder, all subfolders and assets within the folder inherit the restrictions.
- The entire site, all site items inherit the restrictions.

For any item that inherits IP restrictions, you can add additional IP restrictions to further narrow the permitted IP range.

And if a user is denied access to a page, you can redirect them to another page, such as a user-friendly error page.

SEE ALSO:

Adding IP Restrictions in Site.com

Editing, Disabling, and Deleting IP Restrictions in Site.com

Adding IP Restrictions in Site.com

Control site visitors' access to the pages, page templates, folders, and assets in your site by setting the range of permitted IP addresses.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and edit IP restrictions in Site.com:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

1. On the Overview tab, either:
   - Hover over the page, page template, folder, or asset in the All Site Content view and click > **Add IP Restrictions**.
   - Click **Site Configuration** > **IP Restrictions** and click **Add IP Restrictions**.
2. If not already selected, choose the item you want to restrict access to.
3. Enter the first and last IP addresses of the permitted IP range—for example, 112.122.0.1 to 112.122.0.123. To enter a single IP address, just complete the `Start Address` field.

   Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space ::ffff:0:0 to ::ffff:ffff:ffff, where ::ffff:0:0 is 0.0.0.0 and ::ffff:ffff:ffff is 255.255.255.255. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as 255.255.255.255 to ::1:0:0:0 or :: to ::1:0:0:0 are not allowed.
4. Click **Add IP Range** to add additional ranges.
5. To redirect users with an invalid IP address to an alternative page, such as a user-friendly error page, specify the page in Access Denied Page. If you set this in the IP Restrictions view, the page is the default for all IP restrictions unless you override it at the item level.

   You can redirect users to a page in your site or to an external site. Always use a prefix such as `http://` when entering an external URL. If you don't set an access denied page, users see a blank page that displays a default "Access to this resource is denied" message.
6. Click **Save**.

To test the IP restrictions of a page template or site page, click **Preview** when the page is open. When you're happy with your updates, publish the site to enable the restrictions.

Note:

- If an item inherits IP restrictions—for example, an asset in an IP-restricted folder—you can add additional restrictions to further narrow the range. Although the item doesn't display the inherited values anywhere, the inherited IP range values do apply to the child item and only site visitors with valid IP addresses can access it.
- If you select a site page in the Access Denied Page drop-down list, users with an invalid IP address can view that page even if the entire site is restricted.

- Caching is disabled for any item that has IP restrictions. Additionally, if you update the IP restrictions of an asset, folder, or page, the system updates its URL in case proxy servers such as Akamai already cached the item.

SEE ALSO:

Site.comIP Restrictions Overview

Editing, Disabling, and Deleting IP Restrictions in Site.com

Editing, Disabling, and Deleting IP Restrictions in Site.com

After you create an IP restriction for a Site.com site, you can edit the address range, briefly disable the IP restriction (say, to allow temporary access to a page), or delete the restriction entirely.

To view all the IP restrictions on a site, open the IP Restrictions view under Site Configuration (on the Overview tab). Alternatively, to view the IP restriction on a single item, hover over it in the All Site Content view and click > **Edit IP Restrictions**.

When either the IP Restrictions section or the IP restrictions dialog box is open, you can:

- Edit the range of IP addresses. In the IP Restrictions view, double-click the IP address to edit the values inline.
- Disable an IP restriction by deactivating it. You can reactivate the restriction at any time. If you deactivate an IP restriction on an item that has several restrictions, the item's other restrictions are also deactivated.
- Delete an IP restriction. This removes the restriction entirely and allows all site visitors to access the item.

After you update the site's IP restrictions, publish the site to enable your changes.

SEE ALSO:

Site.comIP Restrictions Overview

Adding IP Restrictions in Site.com

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To add and edit IP restrictions in Site.com:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Managing Domains and Live Sites

Managing Domains in Site.com

Before you can publish your site to the Internet, you must set the site's domain information.

To add a domain to the site's domains list, Salesforce must verify that you own the domain name. The verification method you choose depends on whether the domain name is currently in use.

- Add domain names using CNAME records if you're creating a new domain name or using a domain name that's not currently in use.
- Add domain names using TXT records if the domain name is already in use on another website and you want to reuse it. The existing website must also use domain name system (DNS) A records.

SEE ALSO:

Publishing Site Changes

Taking a Site Offline

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Adding Domains Using CNAME Records in Site.com

To publish a site to the Internet, you must first set its domain information. Add a domain name using a CNAME record if you're creating a new domain name, or if you're adding a domain name that's currently not in use.

If you don't already have a branded, custom Web address, such as `www.mycompany.com`, create one by registering through a domain name registrar.

Each domain name in Site.com must be unique. Salesforce verifies your domain name when you add it to the site's list of domains, and again when you publish the site.

Tip: It can take up to 48 hours for domain changes to become available on the Internet. However, you can reduce this time by lowering the time to live (TTL) value in the account management settings of your DNS provider. Because the current remaining time must expire before your new setting takes effect, you should update this value a few days prior to going live.

You can also add a custom path to your domain name to create a custom URL. A custom URL consists of the domain and a custom path. The same path name may be used on more than one domain, but it can't be used more than once within the same domain. When adding a domain, the / is required for the path. It indicates the root. You can add an additional custom name after the /, but you must at least use the / to indicate the root. For example, if the domain name is `https://oursite.com` and the path is `/products`, the site URL is `https://oursite.com/products`. If you added the custom URL to the root, the URL would be `https://oursite.com`.

Note: You can also manage domains and paths in Setup using Domain Management.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

1. In the account management settings of your DNS provider, create a CNAME record. CNAME records must include your domain name, your 18–character organization ID, and the suffix live.siteforce.com. For example, if your domain name is www.mycompany.com and your organization ID is 00dx00000000001aaa, then the CNAME must be `www.mycompany.com.00dx00000000001aaa.live.siteforce.com`. You can find the organization ID on the new domain page in Domain Management within Setup.
2. When your CNAME record is available on the Internet, open Site.com Studio and click **Site Configuration** > **Domains** on the Overview tab.
3. Enter the domain name in the text box provided—for example, `www.mycompany.com`—.
4. Add an optional path name.
5. click Add.
6. Repeat for any additional domain names. For example, you might want to include common misspellings of the domain name in case users make typing mistakes. You must create a corresponding unique CNAME record for each domain name. You can add up to ten domain names.
7. When you're ready to go live, publish the site.

Note:

- If you have an MX record set up for your domain's mail service, the domain assigned to the MX record can't be the same as the domain you assign to your CNAME record. If you create a CNAME record using the same domain as your MX record, your mail service will be disabled.

For example, if you have `mydomain.com` assigned to your MX record, and you want to use it for your CNAME record, we recommend assigning `www.mydomain.com` to your CNAME record instead, and then working with your DNS provider to redirect `mydomain.com` to `www.mydomain.com`.

- When you update existing domain information, you must publish your changes for them to take effect. If you see a message that there are no changes to publish, first update a page in your site and then publish your changes.

SEE ALSO:

Adding Domains Using TXT Records in Site.com

Managing Domains in Site.com

Adding Domains Using TXT Records in Site.com

To publish a site to the Internet, you must first set its domain information. Add a domain name using a TXT record if you're reusing a domain name that's currently in use. For example, if you're migrating from an existing site to a new Site.com site, you can transition seamlessly to the new site when you're ready to go live. This method is suitable only for existing sites that use DNS A records.

Each domain name in Site.com must be unique. Salesforce verifies your domain name when you add it to the site's list of domains, and again when you publish the site.

Tip: It can take up to 48 hours for domain changes to become available on the Internet. However, you can reduce this time by lowering the time to live (TTL) value in the account management settings of your DNS provider. Because the current remaining time must expire before your new setting takes effect, you should update this value a few days prior to going live.

1. In the account management settings of your DNS provider, create a TXT record that contains your organization's ID. To find the ID, open Site.com Studio, click **Site Configuration** > **Domains** on the Overview tab, and copy the organization ID displayed there.

   Contact your DNS provider if you're unsure how to create a TXT record.

2. When your TXT record is available on the Internet, enter the domain name in the text box provided in the Domains view—for example, `www.mycompany.com`—and click **Add**.
3. Repeat for any additional domain names. For example, you might want to include common misspellings of the domain name in case users make typing mistakes.

   You can add up to ten domain names.

4. Publish your Site.com site when it's ready. It won't yet be live on the Internet.
5. To go live, in your DNS account management settings:
   a. Create a CNAME that meets the following criteria.CNAME records must include your domain name, your 18–character organization ID, and the suffix live.siteforce.com. For example, if your domain name is www.mycompany.com and your organization ID is 00dx00000000001aaa, then the CNAME must be *`www.mycompany.com.00dx00000000001aaa`*`.live.siteforce.com`. You can find the organization ID on the new domain page in Domain Management within Setup.

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

b. Delete the old A record, and also the TXT record you created in step 1.

SEE ALSO:

Adding Domains Using CNAME Records in Site.com

Managing Domains in Site.com

Publishing and Managing Live Sites

When you *publish* a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

Before you publish your site for the first time, you must set the site's domain information. If you don't set up your domain, you'll be prompted to do so during the publishing process. After your domain is set up, you can publish your entire site or just parts of it.

When working with a site, you can:

- Publish site changes.
- Review the change history by clicking > **View Details** on the Overview tab.
- Take the site offline to remove it from public view.

Note: You can't publish your site from sandbox.

SEE ALSO:

Using Site.com Studio as a Site Administrator or Designer

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Publishing Site Changes

When you *publish* a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

Before you publish your site for the first time, you must set the site's domain information. If you don't set up your domain, you'll be prompted to do so during the publishing process.

You can publish your entire site or just specific items.

1. Click **Publish Changes...**.
2. To publish:
   - All recent changes, ensure **Site-wide changes** is selected.
   - Specific items, select **Only selected items** and select the relevant items.

   Note: If you select an item that has dependencies, the dependent items are also selected. For example, if you select a page, but you changed the page and the style sheet that it relies on, the style sheet is also selected. Click **View** to see the list of dependencies. You can't deselect an item if it's a critical dependency for another selected item.
3. Click **Next** and then review the list of items to be published.
4. Click **Next** and then add a publishing note, if required. The note appears in the Description column of the Change History view.
5. Click **Publish**.

You'll receive an email notification when your site changes go live.

Note: You can't publish your site from sandbox.

SEE ALSO:

Publishing and Managing Live Sites

Taking a Site Offline

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Taking a Site Offline

1. On the Overview tab, click **Site Configuration** > **Domains**.
2. Delete all listed domains.
   When you remove the last domain, the site goes offline immediately. Site visitors see a "Server not found" message or similar in their browser when they attempt to view the site.
3. In the popup window that appears:
   - Click **OK** to unpublish the site. Unpublishing the site releases the Site.com Published Site license associated with it, so you can use it to publish another site.
   - Click **Cancel** if you don't want to release the Site.com Published Site license—for example, if you plan to make the site publicly available again at a later date.

     If you change your mind after clicking **Cancel**, you can release the license by clicking **Unpublish** on the Overview tab. After a site is unpublished, the **Unpublish** button is replaced with the **Publish** button.
4. If you don't plan to reuse the domain with Site.com, remove any CNAME records that point to *your_domain_name*`.orgID.live.siteforce.com` in the account management settings of your DNS provider.

After you unpublish a site, you can delete it from the list of sites on the Site.com tab.

SEE ALSO:

Managing Domains in Site.com

Publishing Site Changes

EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS

To manage domains and publish Site.com sites:

- `Site.com Publisher User` field enabled on the user detail page

  AND

  Site administrator role assigned at the site level

Force.com Sites

Force.com Sites Overview

Salesforce organizations contain valuable information about partners, solutions, products, users, ideas, and other business data. Some of this information would be useful to people outside your organization, but only users with the right access and permissions can view and use it. In the past, to make this data available to the general public, you had to set up a Web server, create custom Web pages (JSP, PHP, or other), and perform API integration between your site and your organization. Additionally, if you wanted to collect information using a Web form, you had to program your pages to perform data validation.

With Force.com sites, you no longer have to do any of those things. Force.com Sites enables you to create public websites and applications that are directly integrated with your Salesforce organization—without requiring users to log in with a username and password. You can publicly expose any information stored in your organization through a branded URL of your choice. You can also make the site's pages match the look and feel of your company's brand. Because sites are hosted on Force.com servers, there are no data integration issues. And because sites are built on native Visualforce pages, data validation on collected information is performed automatically. You can also enable users to register for or log in to an associated portal seamlessly from your public site.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Note: Force.com Sites is subject to these additional Terms of Use.

For information on Site.com, which is a Web content management system (CMS) that makes it easy to build dynamic, data-driven Web pages and edit content in real time, see Site.com Overview on page 641.

The following examples illustrate a few ways that you can use sites:

- Create an ideas site—Use sites to host a public community forum for sharing and voting on ideas about your company, services, or products. Ideas websites can be made public using sites.
- Publish a support FAQ—Provide helpful information on a public website where customers can search for solutions to their issues.
- Create a store locator tool—Add a public tool to your portal that helps customers find stores in their area.
- Publish an employee directory—Add an employee directory to your company's intranet by creating a site restricted by IP range.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes online.
- Publish a catalog of products—List all of your company's products on a public website, with model numbers, current prices, and product images pulled dynamically from your organization.

Because Force.com sites are served directly from the Salesforce organization, a site's availability is directly related to the organization's availability. During your organization's maintenance window for major releases, your sites will be unavailable; users who try to access a site will see a Force.com-branded maintenance page or your custom Service Not Available Page. It's a good idea to inform your site users of the release maintenance windows and related sites unavailability in advance. You can view specific maintenance windows, listed by instance, at `trust.salesforce.com/trust/status/#maint`.

The Force.com Domain

For each of your sites, you determine the URL of the site by establishing the site's domain name. You can choose one of the following domain options:

- Use your Force.com domain name, which is your unique *subdomain prefix* plus `force.com`. For example, if you choose *`mycompany`* as your subdomain prefix, your domain name would be `http://mycompany.force.com`. The name is case sensitive.

  Note: Your Force.com domain name is used for all the sites that you create. For example, your company could create one public site for partners, another for developers, and a third for support. If your company's domain is `http://mycompany.force.com`, those three sites might have the following URLs:

  - `http://mycompany.force.com/partners`
  - `http://mycompany.force.com/developers`
  - `http://mycompany.force.com/support`

- Create a branded, custom Web address, such as `http://www.mycompanyideas.com`, by registering through a domain name registrar. Create CNAME records to redirect your branded domain and subdomains to your Force.com domain without exposing the `force.com` name in the URL. It can take up to 48 hours for your Force.com domain to become available on the Internet. Custom Web addresses aren't supported for sandbox or Developer Edition organizations.

Note: The format of the secure URLs for your Force.com sites depends on the organization type or Edition. Your unique subdomain prefix is listed first, followed by Edition or environment type, then instance name and the force.com suffix. In the following examples, the subdomain prefix is "mycompany," the sandbox name is "mysandbox," the instance name is "na1," and the sandbox instance name is "cs1":

Organization Type	Secure URL
Developer Edition	`https://mycompany-developer-edition.na1.force.com`
Sandbox	`https://mysandbox-mycompany.cs1.force.com`
Production	`https://mycompany.secure.force.com`

The subdomain prefix for Developer Edition is limited to 22 characters. The secure URL is displayed on the Login Settings page. The URL is case sensitive.

SEE ALSO:

- Setting Up Force.com Sites
- Managing Force.com Sites
- Force.com Sites Limits and Billing
- Administrator setup guide: Force.com Sites Implementation Guide

Managing Force.com Sites

To get started using Force.com sites, from Setup, enter *Sites* in the Quick Find box, then select **Sites**. From this page, you can:

- Register your Force.com domain, if you have not yet done so.
- View all sites hosted on your Force.com domain.
- Click **New** to create a new site.
- Click **Edit** to make changes to an existing site.
- Click **Activate** or **Deactivate** to change the active status for your site.

  Warning: Be careful not to expose any information that you do not want to make public.

- Click the site's label to view that site's details.
- Click the site's URL to open that site in a browser.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

SEE ALSO:

- Setting Up Force.com Sites
- Creating and Editing Force.com Sites
- Configuring Force.com Sites

Registering a Custom Force.com Domain

Your company's Force.com domain is used for all the sites that you create. Even if you register a branded Web address for your site, you must choose a Force.com domain. The branded URL will be redirected to the Force.com domain.

Note: CNAME records must include your domain name, your 18–character organization ID, and the suffix live.siteforce.com. For example, if your domain name is www.mycompany.com and your organization ID is 00dx00000000001aaa, then the CNAME must be *www.mycompany.com.00dx00000000001aaa*.live.siteforce.com. You can find the organization ID on the new domain page in Domain Management within Setup.

To register your Force.com domain:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Enter a unique name for your Force.com domain. A Force.com domain name can contain only alphanumeric characters and hyphens, and must be unique in your organization. It must begin

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

with a letter, not include spaces, not end in a hyphen, and not contain underscores or periods. Salesforce recommends using your company's name or a variation, such as *mycompany*.

Warning: You can't modify your Force.com domain name after you have registered it.

3. Click **Check Availability** to confirm that the domain name you entered is unique. If it isn't unique, you are prompted to change it.
4. Read and accept the Sites Terms of Use by selecting the checkbox.
5. Click **Register My Force.com Domain**. After you accept the Terms of Use and register your Force.com domain, the changes related to site creation are tracked in your organization's Setup Audit Trail and the Site History related list. It may take up to 48 hours for your registration to take effect.

Note: The format of the secure URLs for your Force.com sites depends on the organization type or Edition. Your unique subdomain prefix is listed first, followed by Edition or environment type, then instance name and the force.com suffix. In the following examples, the subdomain prefix is "mycompany," the sandbox name is "mysandbox," the instance name is "na1," and the sandbox instance name is "cs1":

Organization Type	Secure URL
Developer Edition	`https://mycompany-developer-edition.na1.force.com`
Sandbox	`https://mysandbox-mycompany.cs1.force.com`
Production	`https://mycompany.secure.force.com`

The subdomain prefix for Developer Edition is limited to 22 characters. The secure URL is displayed on the Login Settings page. The URL is case sensitive.

SEE ALSO:

Setting Up Force.com Sites

Force.com Sites Considerations

Create a Force.com Site

Setting Up Force.com Sites

Force.com Sites enables you to create public websites and applications that are directly integrated with your Salesforce organization—without requiring users to log in with a username and password.

To set up a Force.com site:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Register a custom Force.com domain for your organization.
3. Create and configure your site.
4. Develop and use Visualforce pages for your site, use or hide standard pages, and customize or replace out-of-box error pages. Associate these pages with your site.
5. Choose a site template. Use the lookup field to find and select a template that you've developed, or use the out-of-box template provided. The site template provides the page layout and stylesheet for your site and overrides any formatting inherited from the associated portal.
6. Enable a portal for login or self-registration, then associate it with your site.
7. Optionally, modify public access settings.
8. When you are ready to make your site public, click **Activate** on the Site Details page. You can also activate your site from the Site Edit and Sites pages.

After you have set up a site, you can:

- Run reports and dashboards on your site.
- Create workflow rules that trigger email alerts when site-related criteria are met.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

To edit public access settings for Force.com sites:
- "Manage Users"

SEE ALSO:

Managing Force.com Sites

Configuring Force.com Sites

Creating and Editing Force.com Sites

After registering your Force.com domain, create a Force.com site using the following steps:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click **New** or click **Edit** for an existing site.
3. On the Site Edit page, define the following:

Field	Description
Site Label	The name of the site as it appears in the user interface.
Site Name	The name used when referencing the site in the SOAP API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

Field	Description
`Site Description`	An optional description of the site.
`Site Contact`	The user who will receive site-related communications from the site's visitors and Salesforce.
`Default Web Address`	The unique Force.com URL for this site. Force.com provides the first part of the URL based on the domain name that you registered; you must add a unique site name to the end. Unless you configure a custom Web address through a domain name registrar, this is the URL that the public will use to access your site.
`Active`	The option that activates your site when you are ready to make it available to the public. You can also activate your site from the Sites and Site Details pages. When the site is inactive, users will see the page specified in the `Inactive Site Home Page` field.
`Active Site Home Page`	The landing page that users are directed to when this site is active. Use the lookup field to find and select the Visualforce page that you developed for your site's home page. Choose the UnderConstruction page if you just want to test your site. For ideas sites, you must use the IdeasHome page and for answers sites you must use the AnswersHome page. The "look and feel" for your Ideas site or Answers site will be inherited from its associated portal unless you select a site template. If you deployed a site before the Summer '10 release, you can set AnswersHome as your default landing page if you create a Visualforce page using `<apex:page action="{!URLFOR('/answers/answersHome.apexp')}"/>`
`Inactive Site Home Page`	The landing page that users are directed to when this site is inactive. Use the lookup field to find and select the page that you want to use. You can, for example, select a page to indicate that the site is under construction or down for maintenance.
`Site Template`	The template that provides the page layout and stylesheet for your site. The site template overrides any formatting inherited from the associated portal. Use the lookup field to find and select a template that you've developed, or use the out-of-box template provided. Note: The site template specified here is used only for Visualforce pages using the `$Site.Template` expression.

`Site Robots.txt`	A file that determines which parts of your public site Web spiders and other Web robots can access. Web robots are often used by search engines to categorize and archive websites. HTML tags are not allowed, as they are not compatible with `robots.txt` spiders. For Visualforce pages, add `contentType="text/plain"` to the `<apex:page>` tag.
	Example disallowing all robots from indexing all pages:
	`<apex:page contentType="text/plain">`
	`User-agent: *  # Applies to all robots`
	`Disallow: /  # Disallow indexing of all pages`
	`</apex:page>`

Field	Description

	Example allowing all robots to index all pages:
	`<apex:page contentType="text/plain">`
	`User-agent: * # Applies to all robots`
	`Disallow: # Allow indexing of all pages`
	`</apex:page>`

Site Favorite Icon	The icon that appears in the browser's address field when visiting the site. Use this field to set the favorite icon for your entire site instead of for each page. Changes will be reflected on your site one day after you make them, due to caching.
Analytics Tracking Code	The tracking code associated with your site. This code can be used by services like Google Analytics to track page request data for your site.
URL Rewriter Class	An Apex class to use for rewriting URLs for your site, from Salesforce URLs to user-friendly URLs. With this class, you can make rules to rewrite site URL requests typed into the address bar, launched from bookmarks, or linked from external websites. You can also create rules to rewrite the URLs for links within site pages.
Enable Feeds	The option that displays the Syndication Feeds related list, where you can create and manage syndication feeds for users on your public sites. This field is visible only if you have the feature enabled for your organization.
Clickjack Protection Level	You can set the clickjack protection for a site to one of these levels: • Allow framing by any page (no protection) • Allow framing by the same origin only (recommended) • Don't allow framing by any page (most protection) Note: Same-origin framing allows the site's page to be framed only by pages on the same domain name and protocol security. Note: Salesforce Communities have two clickjack protection parts—one for the Force.com Communities site which is set from the Force.com site detail page and another for the Site.com Communities site which is set from the Site.com configuration page. It's recommended that both are set to the same value.
Require Secure Connections (HTTPS)	When this checkbox is enabled for a Force.com Site, a Site.com site, a Salesforce Communities Force.com Site, or a Salesforce Communities Site.com site, requests that use HTTP are redirected to HTTPS.

4. Click **Save**.

Note: You define Custom URLs from Domain Management in Setup.

SEE ALSO:

Setting Up Force.com Sites

Configuring Force.com Sites

Force.com Sites-Related Apex Methods and Expressions

Configuring Force.com Sites

Once you have created your site, you can configure it on the Site Details page. To access this page, from Setup, enter *Sites* in the Quick Find box, then select **Sites**, and then in the Sites list, click the site name. You can do the following:

- Click **Edit** to make changes to the site.
- Click **Public Access Settings** to view or edit the security settings, including permissions, page layouts, and more.
- Click **Login Settings** to configure the login and registration settings for your site. Built-in login and registration logic allows users to quickly register for, and seamlessly log in to, your portal from your public site.

  Note: For Communities sites, this link opens the Communities Login Page.

- Click **URL Redirects** to see any page redirects you've set up for your site.
- Click **Activate** or **Deactivate** to change the active status for your site.

  Warning: Be careful not to expose any information that you do not want to make public.

- Click **Preview as Admin** to view your site in administrator preview mode. Administrator preview mode shows the errors on each site page in context and in greater detail, so you can more easily troubleshoot site issues.

  Note: This feature does not appear for community organizations.

- Click **Edit** in the Site Visualforce Pages or Site Standard Pages related lists to select the pages available for your site. All pages associated with the site must be enabled.
- Click **Page Assignment** to assign error pages for standard errors, such as "Authorization Required (401)" and "Page Not Found (404)." You can override or edit the default pages that are provided.
- Click **New** in the Feeds related list to create a new syndication feed. You must click **Enable Feeds** in the Site Detail list to display this related list.
- View the 24-Hour Usage History related list to see current bandwidth and service request time usage, the daily limits, and the percentage used.
- View the Site History related list to see the configuration changes that have been tracked for your site.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

To edit public access settings for Force.com sites:
- "Manage Users"

SEE ALSO:

Assigning Force.com Site Error Pages

Public Access Settings for Force.com Sites

Managing Force.com Site Login and Registration Settings

Defining Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Force.com sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

Define a syndication feed, including what records are returned, and which data from the records is displayed:

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

Create, edit, or delete a feed definition:
- "Modify All Data"

Subscribe to a feed
- No special user permission required

Name

A descriptive name for this feed, which distinguishes it from other feeds you may create. Use only letters, numbers, or the underscore character "_". Do not use more than one underscore character in a row.

Description

Describe the feed. For example, "Account first name, last name, and region for the last ten accounts created or edited."

Query

The SOQL query that defines which records are returned to the syndication feed. To ensure fast performance, some limitations on the SOQL query are imposed:

- If the SOQL query does not specify a limit, then no more than 20 records are returned.
- Query limits can't exceed 200 results. If you make a query with a limit beyond this number, only the first 200 records are returned.
- If the SOQL query does not have an ORDER BY value specified, records are ordered by the `LastModifiedDate` value if there is one, or by `SystemModstamp` value if not
- `COUNT` is not supported.
- *Aggregate* queries are not supported. For example, this query cannot be used in a syndication feeds SOQL query:

```
SELECT Name, (SELECT CreatedBy.Name FROM Notes) FROM Account
```

- You can use *bind variables*, a variable whose value is supplied in the URL. For more information, see Using Bind Variables for Queries and Mapping on page 823.

Note: The guest user must have appropriate sharing and field-level security access or you cannot save your query, because the Force.com platform verifies access and sharing before saving.

Mapping

Because syndication feeds use the ATOM web publishing protocol, you must provide a mapping value that matches objects and fields on the returned records with ATOM constructs. Note that all values must be string literals. For more information about mapping elements, see ATOM-Object Mapping on page 822.

You can use *bind variables*, a variable whose value is supplied in the URL. For more information, see Using Bind Variables for Queries and Mapping on page 823.

Max Cache Age Seconds

Because many users may access a feed at the same time, Salesforce caches the feed data, for 3600 seconds by default. You can modify this to a minimum of 300 seconds, or for as long as you wish. Query results that are older than the time specified here are dropped, and a new query is run on the next request for that information, that is, the next time a user opens a page containing a feed that they have subscribed to.

Active

Select this checkbox to make the feed available for subscription. Once a feed is active, users have the option of subscribing to it.

ATOM-Object Mapping

You must specify a mapping in the syndication feed definition. The mapping relates ATOM constructs such as entry title to the corresponding value in a record, for example, "Account Name." A full set of mappings represents a news feed, and the query represents the content of each news item in a feed. For example, Lead records created today or Contacts with updated Account information.

A *feed element* is the envelope for each part of a news item, and an *entry element* is the contents of the envelope.

Mapping also allows you to apply short labels to various fields for display purposes.

The following table lists each ATOM and object element and explains what values should be provided:

Feed Element	Entry Element	Description
`fa`		Required only if `ea` (entry author) is not specified. Feed author. For example, `fa:"Acme Feed Author Admin Mary"` shows the feed author as Admin Mary.
`fid`		Optional (because default value is supplied). Id of the feed. By default, this value is the public site URL. If you specify a value, it must be a valid internationalized resource identifier (*IRI*). An IRI is a URL generalized to allow the use of Unicode.
`fl`		Optional (because default value is supplied). Feed link. For example, `fl:"http://www.salesforce.com"`. News readers usually interpret this element by linking the feed title to this URL.
`fst`		Optional. Feed subtitle. For example, `&map=ft:"Newest Opportunities",fst:"Western Division"` shows the feed title Newest Opportunities and subtitle Western Division.
`ft`		Required. Feed title. For example, `ft:"Newest Opportunities"`.
	`ea`	Required only if `fa` (feed author) is not specified. Entry author. For example, `ea:"Account created by: " + Account.CreatedBy .`
	`ec`	Required. Entry content. For example,`ec:"description for " Name " " Description` shows the value of the `Name` field with additional text. The output of a feed for this example resembles the following: `description for Ajax Industries Description`
	`ect`	Optional. Entry content of type `text`, `html`, or `xhtml`. For example, `ect: html` for HTML content. Default is `text`.
	`el`	Optional. Entry link. Must be a valid URI. This value is usually a link to another representation of the content for the entry. For example, the link could be to a record in the Salesforce user interface. News readers usually interpret this element by linking the entry title to this URL For example, `el:"Account.URl"`.
	`es`	Optional. Entry summary. An optional summary of the entry content. For example, `et: Account.Name, es: Account.Name + "'s account number, website, and description", ec: Account.AccountNumber + " " + Account.Website + " " + Account.Description` If not specified, news readers display the content defined using the `ec` element.

Feed Element	Entry Element	Description
	est	Optional. Entry summary of type text, html, or xhtml. For example, est: html for HTML content. Default is text. Do not specify a value unless es has been specified.
	et	Required. Entry title, a field name. For example, et:Name.
	eu	Optional. By default, the required ATOM element <updated> value is automatically provided by the LastModifedDate of the main entity being queried; usually the object in the main FROM clause of the SOQL query. This value indicates the last time an entry or feed was modified. If you wish to change this default behavior, you can specify a different object or field's LastModifedDate be used. For example: • Query: SELECT Id, Name, MyDate__c FROM AccountMapping Parameter: eu: MyDate__c • Query: SELECT Id, Lastname, Account.LastMOdifiedDate FROM ContactMapping Parameter: eu: Account.LastModifiedDate

The following example shows a valid mapping values for a syndication feed:

```
ft: "Harry Potter", et: Name, ec: "description for " Name "<br>" Description, el: "/" Id,
 ect: html
```

Feeds are displayed in the guest user context of the public site where they are displayed. Therefore, if you have used custom labels to manage internationalization, and specified labels in your mapping, users see those labels displayed in the language of the guest user. For more information, see Custom Labels and Feeds on page 824.

You can only use string literals in feed mapping. You cannot use, for example, date literals such as TODAY or LAST_WEEK.

After you have defined a feed, you should test it, and then make the feed active by selecting the Active checkbox as described above. For more information about testing, see Testing Syndication Feeds on page 827.

Using Bind Variables for Queries and Mapping

You can use *bind variables* in the WHERE clause of the SOQL query. Bind variables must be in the following form:

```
{!var_name}
```

The following query uses a bind variable named accountID.

```
SELECT Name, Description
FROM Account
WHERE Id = {!accountID}
```

Note that this is not the literal name of the field, but an arbitrary name. At run time, the value for accountID is passed to the query from the URL. This feed is accessed from the following URL and passes in the account ID as part of the query string parameter:

```
site_URL/services/xml/My'Account'Feed?accountId=0013000000BmP4x
```

You can also use bind variables for mapping values.

The following implementation details apply to the use of bind variables for queries:

- You cannot use more than 20 bind variables in a feed definition, queries and mapping combined.
- The bind variable name cannot be more than 100 characters.

- You can use a bind variable only on the right side of a filter operation to represent part of a string. Because it represents part of a string, it must be in quotes. For example, the following is a valid query:

  ```
  SELECT Id, Name FROM Account WHERE Name = '{!myBindVariable}'
  ```

  The following queries are not valid, because the bind variable is not in a valid position, and is not functioning as the variable for a literal string:

  ```
  SELECT Id, Name FROM Account WHERE {!myFieldName} = 'Joe'
  SELECT Id, {!myFieldName} FROM Account WHERE IsDeleted = false
  ```

- You cannot use a bind variable to represent a field name. This means a bind variable cannot be use on the left side of a filter operation.
- You cannot use a bind variable to change the meaning or structure of a query for security reasons. Any special characters you specify in the bind replacement value are interpreted as literal characters when the query is evaluated.

Custom Labels and Feeds

For feeds that need to be localized into different languages, you can use custom labels to define the string in multiple languages. Then in the mapping definition, you simply refer to the custom label. When a request comes in, the custom label inspects the guest user language and returns the translated text, which is used in the mapping.

Custom labels can be specified in a field with the following syntax:

```
map_element_name: "{!$LABEL.custom_label_name}"
```

Use the following procedure to specify a custom label in a feed:

1. From Setup, enter *Custom Labels* in the Quick Find box, then select **Custom Labels**. You may wish to name the custom label after the mapping element that takes its value, for example `feedTitle` for the `ft` element.
2. Enter the values for all supported languages.
3. Specify the custom label in the feed mapping.

For example, assume that you create a feed containing information on all the houses your company is trying to sell. For English users, the title of the feed should be "The Houses," but for Spanish users, the title of the feed should be "Las Casas." You would create a custom label, for example, `feedTitle`. In English, its value is "The Houses," and the Spanish value is "Las Casas." Then, in the feed mapping definition, specify the following for the feed title `fc:`:

```
ft: "{!$LABEL.feedTitle}"
```

Visualforce and Feeds

To add a feed to a Visualforce page, use the Visualforce standard HTML features. For example, assuming the Visualforce page is located in the base directory of the site, it can contain a tag like the following:

```
<A HREF=""/services/xml/theFeedName">My feed</A>
```

The text `My feed` links to the feed.

If you want to link the feed from an image, include an inline image tag similar to the following:

```
<A HREF="/services/xml/theFeedName"><img src="feed.gif"></A>
```

You must upload your own image.

To add the icon to the address bar, add the link tag to the `<head>` tag of the Visualforce page:

```
<link href='URI of feed'
   type='application/x.atom+xml'
   rel='feed'
   title='A nice descriptive title'/>
```

SEE ALSO:

About Syndication Feeds

Testing Syndication Feeds

About Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Force.com sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

Create, edit, or delete a feed definition:

- "Modify All Data"

Subscribe to a feed

- No special user permission required

Validating Feeds Security

When a user subscribes to a feed, the information is run in a guest user context. This means that you must ensure the guest user has access to all and only the records appropriate for a guest use before defining any queries.

To validate feeds security:

- Edit the public access setting for the site to make sure the guest user has the correct object permissions and field-level security settings.
- Create sharing rules to control which records the guest user has access to.

After adjusting public access and field-level security settings to ensure the objects you wish to include in a feed are available to the guest user, perform any of the following feeds-related tasks:

- To create a new feed, click **New**.
- To view the definition of an existing feed, click the feed name.
- To edit an existing feed, click **Edit**.
- To delete an existing feed, click **Delete**.
- To test the validity of a feed, click **Run Test**. If any errors exist in the query definition or mapping, error messages are displayed.

SEE ALSO:

Defining Syndication Feeds

Testing Syndication Feeds

Viewing Syndication Feeds

View the syndication feed definition, including what records are returned, and which data from the records is displayed:

Name
A descriptive name for this feed, which distinguishes it from other feeds you may create.

Description
Describes the feed. For example, "Account first name, last name, and region for the last ten accounts created or edited."

Query
The SOQL query that defines which records are returned to the syndication feed. To ensure fast performance, some limitations on the SOQL query are imposed. For more information, see Defining Syndication Feeds on page 821.

Mapping
Because syndication feeds use the ATOM web publishing protocol, you must provide a mapping value that matches objects and fields on the returned records with ATOM constructs. Note that all values must be string literals. For more information about mapping elements, see Defining Syndication Feeds on page 821.

Max Cache Age Seconds
Because many users may access a feed at the same time, Salesforce caches the feed data, for 3600 seconds by default. This value can be a minimum of 300 seconds, or for as long as you wish. Query results that are older than the time specified here are dropped, and a new query is run on the next request for that information, that is, the next time a user opens a page containing a feed that they have subscribed to.

Active
This checkbox indicates whether the feed is available for subscription. Once a feed is active, users have the option of subscribing to it.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

Create, edit, or delete a feed definition:
- "Modify All Data"

Subscribe to a feed
- No special user permission required

SEE ALSO:

About Syndication Feeds
Testing Syndication Feeds

Testing Syndication Feeds

Test the feed definition you have created before enabling it for customers.

To test a feed:

1. After creating a feed, from Setup, enter *Sites* in the Quick Find box, then select **Sites** and click the site for which you've defined the feed. Alternatively, you can navigate to the feed detail page any number of ways, including clicking the feed name from the site detail page.
2. Click **Preview** for the feed you wish to test.
3. If one or more bind variables have been used in the feed, a dialog appears. Enter a test value for each bind variable.
4. A dialog appears allowing you to create a bookmark for the feed with the bind variable values you specified. You can save the bookmark, or cancel the dialog.
5. The values returned by your feed are displayed. Verify that the results are what you expected.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

Create, edit, or delete a feed definition:
- "Modify All Data"

Subscribe to a feed
- No special user permission required

SEE ALSO:

Defining Syndication Feeds
About Syndication Feeds

Managing Force.com Site Login and Registration Settings

Use login settings if you want to let users register for and log in to your portal from your public Force.com site. For example, users browsing through an ideas site can register and login directly from that site, and as authenticated users, they can then vote, add comments, and participate in the ideas community. When users successfully log in, they leave the public site and enter the associated portal seamlessly.

Note: Only Customer Portals can be used for self-registration. Partner portals do not support self-registration.

The Authenticated Website high-volume portal user license is specifically designed to be used with Force.com sites. Because it's designed for high volumes, it should be a cost-effective option to use with Force.com sites.

Force.com sites provides built-in registration and login logic. Default Force.com-branded Visualforce pages are associated with registration, login, forgot password, and password changes. You can modify these pages or replace them with your own.

The following login, and registration pages are provided by default:

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

Page Name	Description
SiteLogin	Default login page. Used to log users in to the associated portal from your Force.com site.
SiteRegister	Default registration page. Used to register new users for the associated Customer Portal from your Force.com site.
SiteRegisterConfirm	Default registration confirmation page. The page that users see on successful registration to the associated Customer Portal.

Note: The login and registration forms must be secure. Set the `forceSSL` attribute to `true` for these forms. However, Salesforce recommends that you set `forceSSL` to `false` for forms accessed by users who have already been authenticated, such as portal users. Since the `forceSSL` attribute forces a redirect to a secure URL, authenticated users would encounter an error.

The built-in login process:

- Checks to see whether the site is enabled for logins
- Checks to see whether the user is a valid user for the site
- Allows users to reset expired passwords

The built-in registration process:

- Checks new user information against existing users for the Customer Portal associated with the site
- Checks to see if a contact already exists for the new user
- Creates a new contact, if one doesn't already exist, and associates it with the account for the site.

  Important: You must update the SiteRegisterController with the Account ID using the following steps:

  1. From Setup, enter *Apex Classes* in the `Quick Find` box, then select **Apex Classes**.
  2. Click **Edit** next to SiteRegisterController.
  3. Find the `private static Id PORTAL_ACCOUNT_ID = '<Account_ID>';` line and insert the ID for the account that you want to associate with new users. The line should look similar to this:

     ```
     private static Id PORTAL_ACCOUNT_ID = '001DooooolQpyk';
     ```

  4. Click **Save**.

- Enables the Customer Portal for the new user and sends an email confirmation message
- Optionally, allows users to create passwords on the registration page, avoiding the standard email confirmation process

Note: You can create and enable a person account as a Customer Portal user using the `createPersonAccountPortalUser` Apex method. Use `createPersonAccountPortalUser` to create a person account using either the default record type defined on the guest user's profile or a specified record type, then enable it for the site's portal. Person Accounts can only be enabled as high-volume portal users.

To enable public login and registration for your portal:

1. From Setup, enter *Sites* in the `Quick Find` box, then select **Sites**.
2. Click the name of the site you want to control.
3. Click **Login Settings**.
4. Click **Edit**.
5. From the Enable Login For list, select a portal to associate with your site. The portal you choose must have the `Login Enabled` option selected. For Customer Portals, you must also select the `Self-Registration Enabled` option. Force.com Sites leverages the following portal settings:
   - `Logout URL` is used if you want to take users to a specific page on logout. If this value is left blank, users are taken to the page specified in the `Active Site Home Page` field for your site.
   - `Lost Password Template` is used for the forgot password process.
   - `Header`, `Footer`, `Logo`, and `Login Message` are used for the look and feel on IdeasHome and AnswersHome pages.
   - For Customer Portals:

- `New User Template` is used on self-registration if a password is not provided.
- `Default New User License`, `Default New User Role`, and `Default New User Profile` are used for self-registration.

6. Select a **Change Password Page**. A default page is provided, but you can select your own page instead, using the lookup field.
7. Select `Require Non-Secure Connections (HTTP)` to override your organization's security settings and exclusively use HTTP when logging in to the portal from your site. If this checkbox is not selected, the `Require Secure Connections (HTTPS)` setting on the Session Settings page in Setup determines the security level.

   The following table illustrates the relationship between these settings.

Site-Level Security: Require Non-Secure Connections (HTTP)	Organization-Level Security: Require Secure Connections (HTTPS)	Description
Not checked	Not checked	• Organization can use either HTTP or HTTPS • Site uses HTTP for the post-login session
Checked	Not checked	• Organization can use either HTTP or HTTPS • Site uses HTTP for the post-login session
Not checked	Checked	• Organization uses only HTTPS • Site uses HTTPS for the post-login session • Upon login, users see the `secure.force.com` domain
Checked	Checked	• Organization uses only HTTPS • Site uses HTTP for the post-login session

   Warning: If the `Require Secure Connections (HTTPS)` checkbox on the Session Settings page is selected, and the `Require Non-Secure Connections (HTTP)` checkbox on the Login Settings page is not selected, users logging in to the associated portal from the site will see the `secure.force.com` domain. For example, if you registered `mycompany.force.com` as your custom domain, the URL changes to `https://mycompany.secure.force.com` upon login.

   Customers using a script to login to sites can use the optional `refURL` URL parameter to retain the custom domain name after login. This parameter has no effect if `Require Non-Secure Connections (HTTP)` has been set for the site or `Require Secure Connections (HTTPS)` has been set for the organization. An example URL using `refURL` is: `http://mysite.secure.force.com/SiteLogin?refURL=http://mysite.com`.

8. The `Secure Web Address` field shows the unique Force.com URL for this site when using SSL.

9. Click **Save**.

You can also enable Sites to use your identity provider for single sign-on.

SEE ALSO:

Configuring Force.com Sites

Associate a Portal with Force.com Sites

Force.com Sites-Related Apex Methods and Expressions

Public Access Settings for Force.com Sites

Public access settings control what public users can do on each Force.com site. To set the public access settings for your site:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click the name of the site you want to control.
3. Click **Public Access Settings** to open the Profile page for your site profile.

This page includes all the functionality for viewing and editing profile permissions and settings, but you can't clone or delete the profile.

In the site profile, you can:

- Set the object permissions for your site. You can grant "Read" and "Create" permissions on all standard objects except products, price books, and ideas; and "Read," "Create," "Edit," and "Delete" on all custom objects. All permissions that aren't set by default must be set manually.

  Warning: We recommend setting the sharing to private for the objects on which you grant "Read" access for your site. This ensures that users accessing your site can view and edit only the data related to your site.

  We also recommend securing the visibility of all list views. Set the visibility of your list views to Visible to certain groups of users, and specify the groups to share to. List views whose visibility is set to Visible to all users may be visible to public users of your site. To share a list view with public users, create a new public group for those users and give them visibility. If the object's sharing is set to private, public users won't be able to see those records, regardless of list view visibility.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

To edit public access settings for Force.com sites:
- "Manage Users"

- Control the visibility of custom apps. If you want to expose a custom app and its associated tabs to public users, make only that app visible and make it the default to avoid exposing other pages. If any of your site pages use standard Salesforce headers, other visible applications may be seen by public users.
- Set the login hours during which users can access the site.
- Restrict the IP address ranges from which you can access the site. Force.com sites ignore company-wide IP range restrictions in order to provide public access; however, you can restrict the IP range here.

  Note: To set restrictions based on IP or login hours, HTTPS is required. You must use the secure URL associated with your Force.com domain to access your site.

  To enforce HTTPS on all Force.com sites pages and allow all IP addresses to access your site, create the following IP ranges: 0.0.0.0 to 255.255.255.255, :: to ::fffe:ffff:ffff, and ::1:0:0:0 to ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff. However, as this may degrade the performance of your site, don't enforce HTTPS unless it is absolutely required. Changing from HTTP to HTTPS doesn't affect logged in users until the next time they log in.

The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` aren't allowed.

- Enable Apex controllers and methods for your site. Controllers and methods that are already associated with your site's Visualforce pages are enabled by default.
- Enable Visualforce pages for your site. Changes made here are reflected on the Site Visualforce Pages related list on the Site Details page, and vice versa.

SEE ALSO:

Configuring Force.com Sites

Force.com Sites-Related Apex Methods and Expressions

Force.com Sites URL Redirects

If you move or reorganize pages on your Force.com site, search engines may have trouble finding the new page locations. To avoid this, set up *site URL redirects* to inform users and search engines that site content has moved.

Consider the following while implementing site URL redirects:

- You can't redirect error pages or CSS files (files with a `.css` extension).
- Each site can have a maximum of 1,000 redirect rules.
- Query parameters in site URL redirects are matched exactly. However, you can't redirect any URLs that include the `lastMod` parameter.
- If you have URL rewriting enabled on your site, it runs after any site page redirects.
- You can redirect a community home page to its companion Site.com home page. To do this, set the Source URL to `/`, which represents the home page for the community, and set the Target URL to `s`, which represents the home page for the Site.com site.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit site URL redirects:
- "Customize Application"

To view site URL redirects:
- "View Setup and Configuration"

To assign a redirect to a site page:

1. From Setup, enter *`Sites`* in the `Quick Find` box, then select **Sites**.
2. Click a site label.
3. Click **URL Redirects** on the site detail page.
4. Specify the former page location in the `Source URL` field. The page location must:
   - Be a relative URL. It can have any valid extension type, such as `.html` or `.php`.
   - Not contain anchors, such as `/siteprefix/page.html`**`#target`**.

   Note: Organizations that use sites with prefixes must manually add the prefix to the Source URL and Target URL fields. Also, if your organization has a root level site and one with a prefix, and you want to redirect a page in your prefixed site but don't include the prefix in the redirect rule, Salesforce will default to looking for the rule in your root site instead, resulting in a 404 error.
5. Specify the `Redirect Type`:
   - Permanent (301)—Select this option if you want users and search engines to update the URL in their systems when visiting the page. Users visiting a page redirected with this type are sent seamlessly to the new page. Using a permanent redirect ensures

that your URLs retain their search engine popularity ratings, and that search engines index the new page and remove the obsolete source URL from their indexes.

- Temporary (302)—Select this option if you want users and search engines to keep using the original URL for the page. Search engines interpret a 302 redirect as one that could change again at any time, and though they index and serve up the content on the new target page, they also keep the source URL in their indexes.

6. Specify the new page location in the `Target URL` field. This can be a relative URL or a fully-qualified URL with an `http://` or `https://` prefix. Unlike source URLs, target URLs can contain anchors.
7. Click **Save**.

The Redirect Rules section displays all URL redirect rules you've created for your site. In this section you can:

- Edit an assigned redirect rule.
- Activate or deactivate a redirect rule.
- Delete a redirect rule.
- Click the Source URL or Target URL column heading to sort the list in ascending or descending order.

SEE ALSO:

Force.com Sites Overview

Associate a Portal with Force.com Sites

You can enable users to register for or log into an associated portal seamlessly from your site.

Note: Only Customer Portals can be used for self-registration. Partner portals do not support self-registration.

The Authenticated Website high-volume portal user license is specifically designed to be used with Force.com sites. Because it's designed for high volumes, it should be a cost-effective option to use with Force.com sites.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

1. Enable the portal for login using the following steps:
   a. From Setup, enter *`Customer Portal Settings`* in the `Quick Find` box, then select **Customer Portal Settings**, or from Setup, enter *`Partners`* in the `Quick Find` box, then select **Settings**.
   b. If you have not enabled your portal, select `Enable Customer Portal` or `Enable Partner Relationship Management` and click **Save**.
   c. Click **Edit** for the portal you want to modify.
   d. Configure the portal as follows:
      i. Select the `Login Enabled` checkbox.
      ii. Select a user for the `Administrator` field.
      iii. Optionally, set the `Logout URL`. If this is not set, users are taken to the site home page on logout.
      iv. Click **Save**.
2. If you are using a Customer Portal and want to allow self-registration, follow these steps:
   a. From Setup, enter *`Customer Portal Settings`* in the `Quick Find` box, then select **Customer Portal Settings**.
   b. Click **Edit** for the portal you want to associate with your Force.com site.

c. Configure the Customer Portal as follows:

i. Select `Self-Registration Enabled`.

ii. Select Customer Portal User for both the `Default New User License` and `Default New User Profile` fields. Depending on your portal license, you may want to select a different profile for the `Default New User Profile` field.

iii. Select User for the `Default New User Role` field.

iv. Click **Save**.

Note: Consider the following when allowing self-registration:

- Sites does not support the use of Person Accounts for self registration.
- On self-registration through a site:
  - Validation rules are enforced on user creation.
  - Validation rules are ignored on contact creation.

3. Associate the site pages with the default portal users:

a. From Setup, enter *`Customer Portal Settings`* in the `Quick Find` box, then select **Customer Portal Settings**, or from Setup, enter *`Partners`* in the `Quick Find` box, then select **Settings**.

b. Click the name of the portal that you want to associate with your site.

c. Click the name of each profile associated with your portal users and do the following:

i. Scroll down to the Enabled Visualforce Page Access section and click **Edit**.

ii. Add the appropriate public site pages to the `Enabled Visualforce Pages` list. This allows portal users with that profile to view these pages.

Note: By default, portal users can see all pages enabled for the associated public site, so you only have to enable the pages that require authentication.

iii. Click **Save**.

4. Associate your site with the login-enabled portal:

a. From Setup, enter *`Sites`* in the `Quick Find` box, then select **Sites**.

b. Click the site label of the site you want to configure.

c. Click **Login Settings**.

d. Click **Edit**.

e. From the `Enable Login For` drop-down list, select the name of the portal where you want to allow login.

f. Select the `Change Password Page`.

g. Click **Save**.

For sites with Ideas, Answers, Chatter Answers, you must make the zone visible in the portal and enable the IdeasHome or AnswersHome page for the site. To associate the zone with one or more portals:

1. From Setup, search for one of the following in the `Quick Find` box.

- **Ideas Zones**
- **Chatter Answers Zones**

- **Answers Zones**

2. Click **Edit** next to the zone you want to make public.
3. From the `Portal` drop-down list, select the portal to use for this zone. You can choose to show the zone in all portals.

Note: For ideas to work with sites, the organization must have an active portal associated with that zone. Otherwise, users will encounter errors.

SEE ALSO:

Configuring Force.com Sites

Managing Force.com Site Login and Registration Settings

Managing Force.com Site Visualforce Pages

Force.com sites uses Visualforce pages for all site and error pages. Sample error pages use the SiteSamples static resource for their stylesheet and images.

Warning: Don't rename or delete SiteSamples or you may get an error.

All pages that you want to expose on a site must be associated with that site. If a page is not listed under Site Visualforce Pages, an authentication or page-not-found error is displayed based on the existence of the page.

To enable Visualforce pages for your site:

1. From Setup, enter *`Sites`* in the `Quick Find` box, then select **Sites**.
2. Click the name of the site you want to modify.
3. Click **Edit** on the Site Visualforce Pages related list.
4. Use the **Add** and **Remove** buttons to enable or disable Visualforce pages for your site.
5. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

Note: If you select a Visualforce page for any of the lookup fields on the Site Detail page, any of the error pages, or the `Change Password Page` under login settings, that page is automatically enabled for your site. If you remove a page from this list, but it is still selected in one of these places, public users can access that page. To completely remove pages from your site, disable them here and make sure they are not selected in any lookup fields for your site.

If you don't want to enable a Visualforce page for your entire site, you can also enable pages for specific profiles.

The My Profile Page

The My Profile page is a Visualforce page associated with a Customer Portal or site user's profile. The My Profile page enables users logged into either your Force.com site, or your Customer Portal from Force.com sites, to update their own contact information. When they make changes to this page, the corresponding portal user and contact records are updated.

The My Profile page can be enabled either for your entire site or in a more restricted way by assigning it to the site guest user profile.

The My Profile page is also delivered as part of the Customer Portal Welcome component on your home page layout.

SEE ALSO:

Managing Force.com Site Standard Pages

Assigning Force.com Site Error Pages

Managing Force.com Site Standard Pages

Force.com sites uses Visualforce pages for all site and error pages. Force.com also provides some standard pages that you can use.

All pages that you want to expose on a site must be enabled for that site. If a page is not listed under Site Standard Pages, Salesforce displays an authorization required error.

To control which standard pages users see for your site:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click the name of the site you want to view.
3. Click **Edit** on the Site Standard Pages related list.
4. Use the **Add** and **Remove** buttons to enable or disable the following standard pages for your site:
   - Home Page—The standard page associated with the Home tab (`/home/home.jsp`).
   - Ideas Pages—The standard pages associated with ideas. If you want to use default ideas pages (for example, IdeasHome), enable these pages.
   - Answers Pages—The standard pages associated with answers. If you want to use default answers pages (for example, AnswersHome), enable these pages.
   - Search Pages—The standard Salesforce search pages. To allow public users to perform standard searches, enable these pages.
   - Lookup Pages—The standard Salesforce lookup pages. These are the popup windows associated with lookup fields on Visualforce pages.
5. Click **Save**.

Note: Make sure to disable any pages that you are not actively using in your site. Otherwise, they may be accessible to public users. Also, make sure to set up private sharing to restrict search and lookup access for public users.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

SEE ALSO:

Managing Force.com Site Visualforce Pages

Assigning Force.com Site Error Pages

Assigning Force.com Site Error Pages

Force.com sites uses Visualforce pages for all site and error pages. Sample error pages use the SiteSamples static resource for their stylesheet and images.

Warning: Don't rename or delete SiteSamples or you may get an error.

To set the error pages for your site:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click the name of the site you want to modify.
3. Click **Page Assignment** on the Error Pages related list.
4. Using the lookup fields, assign a Visualforce page or static resource for each of the standard error conditions listed:
   - `Authorization Required Page`—The page users see when trying to access pages for which they do not have authorization.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

- `Limit Exceeded Page`—The page users see when your site has exceeded its bandwidth limits.
- `Maintenance Page`—The page users see when your site is down for maintenance.
- `Service Not Available Page`—The page users see when Salesforce servers are unavailable. This custom page is rendered from a static resource that you choose. The static resource is uploaded to the cache server when assigned as the Service Not Available page, and when updated after assignment. The custom page is shown for HTTP requests only; caching is not used for HTTPS. Not available for Developer Edition or sandbox organizations.

  If you haven't set a custom page for a site that uses a site prefix, the custom page for the root site is used when servers are unavailable. For example, if `http://mycompany.force.com/sales` doesn't have a custom page, but `http://mycompany.force.com` does, that custom page is used. Otherwise, the Maintenance page is shown by default.

  Note: The static resource:
  - Must be a public .zip file 1 MB or smaller.
  - Must contain a page named `maintenance.html` at the root level of the .zip file. Other resources in the .zip file, such as images or CSS files, can follow any directory structure.
  - Must contain only files that have file extensions.

- `Page Not Found Page`—The page users see when trying to access a page that cannot be found. You can use the action attribute on an `<apex:page>` component to redirect the Page Not Found error page. Using this kind of redirect on any other error pages will redirect users to the Maintenance page.
- `Generic Error Page`—The page users see when encountering generic exceptions.

Note: When using static resources in a custom error page—such as a large image file or a large CSS file contained in a static resource .zip file—each individual resource must be no larger than 50KB. Otherwise, a 404 not found error is returned for that resource.

5. Click **Save**.
6. On the Site Details page, click **Preview** to view the associated page as it would appear in a browser.

Tip: Add the `<site:previewAsAdmin />` component right before the closing `</apex:page>` tag in your custom Visualforce error pages to view detailed site error messages in administrator preview mode.

SEE ALSO:

Managing Force.com Site Visualforce Pages

Managing Force.com Site Standard Pages

Troubleshooting Force.com Sites Using Administrator Preview Mode

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

If you see errors on site pages and can't figure out the cause, use administrator preview mode to look at them in context and in greater detail.

Note: Administrator preview mode is available for all active sites, including those with a branded custom Web address.

To access administrator preview mode:

1. From Setup, enter *Sites* in the `Quick Find` box, then select **Sites**.

2. Click the name of the site you want to preview.
3. In the Site Detail section, click the **Preview as Admin** link. A new browser window opens with a preview of your site, and the enhanced error message appears at the bottom of the page.
4. Click **Logout of Administrator Preview Mode** to clear the administrator cookie and be redirected to the site's home page.

The detailed error messages in administrator preview mode are driven by the `<site:previewAsAdmin />` component in your Visualforce error pages. Starting with Summer '10, new organizations include the `<site:previewAsAdmin />` component by default in standard error pages. You must add the component manually to all custom error pages and pages from older organizations. We recommend that you add it right before the closing `</apex:page>` tag, like this:

```
    <site:previewAsAdmin />
</apex:page>
```

Note: The `<site:previewAsAdmin />` component contains the `<apex:messages />` tag, so if you have that tag elsewhere on your error pages, you will see the detailed error message twice.

SEE ALSO:

Creating and Editing Force.com Sites

Assigning Force.com Site Error Pages

Enable Single Sign-On for Sites

Single sign-on allows users to access all authorized network resources without having to log in separately to each resource. You validate usernames and passwords against your corporate user database or other client application rather than having separate user passwords managed by Salesforce.

You can set up Sites to use SAML single sign-on, so that a customer only has to login once.

To enable single sign-on for Sites using an identity provider:

1. Set up either a Customer Portal or a partner portal.
2. Set up your Sites.
3. If you are using the Customer Portal, set up your Sites to use Customer Portal.
4. In addition to the SAML sign-on information that must be gathered and shared with your identity provider, you must supply your information provider with the following:
   - Organization ID
   - Portal ID
   - Sites URL

   The following must be added as attributes to the SAML assertion sent from your identity provider:
   - `organization_id`
   - `portal_id`
   - `siteurl`

   To find these values:

   a. From Setup, enter *`Company Information`* in the `Quick Find` box, then select **Company Information** and copy the ID located in the `Salesforce Organization ID`.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

b. For the Customer Portal, from Setup, enter *Customer Portal Settings* in the Quick Find box, then select **Customer Portal Settings**. Next, click the name of the Customer Portal, and then copy the ID located in the Portal ID.

For partner portals, from Setup, enter *Partners* in the Quick Find box, then select **Settings**. Next, click the name of the partner portal, and copy the ID located in the Salesforce Portal ID.

c. From Setup, enter *Custom URLs* in the Quick Find box, then select **Custom URLs**.

Service Provider Initiated SAML for Sites

Salesforce automatically provides single sign-on for Sites using SAML for when your company's identity provider initiates login.

If you wanted to use SAML for Sites for when a service provider initiates sign-on, you must first create a Visualforce page that provides a redirect to your server. The following is an example:

```
<apex:page showHeader="false" sidebar="false">
 <script>
     var PingSpURL =
"https://my.pingserver.com:9031/idp/startSSO.ping?PartnerSpId=salesforce.com.sp";
     var siteLoginPage =
"&TargetResource={!$Site.CurrentSiteUrl}siteLogin?startUrl={!$Site.OriginalUrl}";
     window.location = PingSpURL+siteLoginPage;
 </script>
</apex:page>
```

Caching Force.com Sites Pages

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Force.com sites enforces two 24-hour rolling limits—bandwidth and service request time—which vary by organization type. Though the limits are high for active production organizations, your site could exceed the limit due to heavy traffic or pages that consume lots of bandwidth or processing time. "Rolling 24-hour period" refers to the 24 hours immediately preceding the current time.

Sites provide caching options that allow you to leverage the resources of our Content Delivery Network (CDN) partner to improve page load times and site performance, as well as help you avoid reaching bandwidth or service request time limits. Sites allows you to set the cache duration for each of your site pages and optimize content delivery to your end users.

Control the caching behavior for your site by setting the Boolean `cache` attribute and integer `expires` attribute on each Visualforce page. By default, pages that do not have the `cache` attribute set are cached for ten minutes (600 seconds).

For example, a page whose cache is set to expire in 15 minutes looks like this:

```
<apex:page cache="true" expires="900">
```

Note: The CDN is only available for active production organizations. It is not available for sandbox or Developer Edition organizations.

To protect the integrity of sensitive information, SSL sessions and pages requested after authentication are not cached via the CDN.

Attachments to public pages accessed through sites are automatically cached for 10 minutes via the CDN.

Other factors also affect caching behavior, such as the type of user accessing the page, whether the request comes from the caching server, and whether the site is login-enabled. The following tables summarize the caching behavior for each of these cases.

For Requests Coming from the Caching Server

If the site is login-enabled:

Cache Attribute	Caching Behavior	Caching Location
Set to `TRUE`	Cache expires according to user-set value	Caching server only
Not Set	Cache expires in ten minutes	Caching server only
Set to `FALSE`	Not cached	None

If the site is **not** login-enabled:

Cache Attribute	Caching Behavior	Caching Location
Set to `TRUE`	Cache expires according to user-set value	Both caching server and browser
Not Set	Cache expires in ten minutes	Both caching server and browser
Set to `FALSE`	Not cached	None

For Requests Not Coming from the Caching Server

If the site is login-enabled:

Cache Attribute	Caching Behavior	Caching Location
Set to `TRUE`	Not cached	None
Not Set	Not cached	None
Set to `FALSE`	Not cached	None

If the site is **not** login-enabled:

Cache Attribute	Caching Behavior	Caching Location
Set to `TRUE`	Cache expires according to user-set value	Browser only
Not Set	Cache expires in ten minutes	Browser only
Set to `FALSE`	Not cached	None

SEE ALSO:

Viewing 24-Hour Force.com Sites Usage History
Force.com Sites Limits and Billing

Using Workflow for Force.com Sites

With workflow for sites, you can create workflow rules that trigger email alerts when certain site-related criteria are met. For example, create a rule that sends your site administrator an email alert when bandwidth usage reaches 80% of the daily bandwidth limit or when a site's status changes from the active state.

Setting up the right site usage workflow rules can help you keep your sites from exceeding rolling 24-hour limits for bandwidth and service request time, and monthly limits for page views and logins. Workflow rules that are created on the Site object are evaluated every hour for all sites within the organization, unless your criteria is limited to certain sites. Workflow rules that are created on the Organization and User License objects are evaluated every three hours.

Only email alert actions are supported for site usage workflow rules. Other workflow actions, such as field updates, tasks, and outbound messages, are not available.

To create a site usage workflow rule, create a workflow rule with the following settings:

Select Object

When selecting an object, choose one of the following:

- `Organization` (for monthly page views allowed and monthly page views used fields)
- `Site` (for site detail, daily bandwidth and request time, monthly page views allowed, and other fields)
- `User License` (for the monthly logins allowed and monthly logins used fields)

The Organization and Site objects are only available if Force.com sites is enabled for your organization. The User License object isn't dependent on sites, and is only available if you have Customer Portals or partner portals enabled for your organization.

Rule Criteria

Enter your rule criteria by doing one of the following:

- Choose `criteria are met` and select the filter criteria that a site must meet to trigger the rule. For example, to trigger the rule every time the active status changes for a site within your organization, set the filter to *`Site Status not equal to Active`*. To add more rows or to set up Boolean conditions, click **Add Filter Logic**.
- Choose `formula evaluates to true` and enter a formula. The rule is triggered if the formula returns "True."

  For example, the following formula triggers the rule when bandwidth usage reaches 80% of the daily bandwidth limit:

  ```
  DailyBandwidthUsed >= 0.8 * DailyBandwidthLimit
  ```

  In the next example, the formula triggers the rule when time usage reaches 80% of the daily time limit:

  ```
  DailyRequestTimeUsed >= 0.8* DailyRequestTimeLimit
  ```

Workflow actions

When adding workflow actions, click **Add Workflow Action** and choose `New Email Alert` on page 616 or `Select Existing Action`.

Tip: To create custom email templates using Site merge fields, from Setup, enter *`Email Templates`* in the `Quick Find` box, then select **Email Templates**, then **New Template**. When configuring the template, use Site merge fields by selecting **Site Fields** in the `Select Field Type` drop-down list of the Available Merge Fields section.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change workflow rules:
- "Customize Application"

To create and edit Force.com sites:
- "Customize Application"

SEE ALSO:

Force.com Sites Overview

Tracking Your Force.com Site with Google Analytics

Force.com sites provides out-of-the-box integration with Google Analytics. Use Google Analytics to track the usage of your sites and site pages, including number of visits, number of page views, average time spent on site, and more.

Note: The `<site:googleAnalyticsTracking/>` component only works on pages used in a Force.com site. Sites must be enabled for your organization and the `Analytics Tracking Code` field must be populated. To get a tracking code, go to the Google Analytics website.

To track a site using Google Analytics:

1. Sign up for an account at Google Analytics.
2. Add a new profile in Google Analytics and enter the domain or full URL for the site you want to track.
3. Copy the `Web Property ID` from Google's tracking status information and paste it into the `Analytics Tracking Code` field on the Site Edit page for the site you want to track. The Web property ID starts with the letters UA followed by your account and profile numbers. For example, UA-9049246-2.
4. Click **Save**.
5. To track the Visualforce pages associated with your site, enter the following tag in the site template for those pages, or in the individual pages themselves:

   ```
   <site:googleAnalyticsTracking/>
   ```

   Pages that don't contain the tag and aren't associated with a site template that contains the tag won't be tracked. The default site template already contains the tag, so all pages using that template will be tracked—including certain default pages.

   Note: Google recommends adding the component at the bottom of the page to avoid increasing page load time.

6. Go to the Google Analytics site and follow their instructions for completing the process. After signing up, it may take up to 24 hours to see initial tracking results in Google Analytics.

Tip: To track multiple sites separately, create separate profiles using the full site URLs and enter a different Web property ID in the `Analytics Tracking Code` field for each site.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

SEE ALSO:

Managing Force.com Sites

Creating and Editing Force.com Sites

Viewing 24-Hour Force.com Sites Usage History

Reporting on Force.com Sites

Viewing Force.com Site History

To view the site history:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click the name of the site you want to view.
3. View the Site History related list.

The Site History related list tracks and displays the changes made to your site. All of the following events are tracked in the site history, along with the user who made the change and the time it occurred:

Event	Description
Site Creation	Logs when each site was created.
Site Detail Changes	Changes to the following site values are logged: • Site Label • Site Name • Site Description • Site Contact • Default Web Address • Custom Web Address • Active Status • Active Site Home Page • Inactive Site Home Page • Site Template • Site Robots.txt • Site Favorite Icon • Analytics Tracking Code • Enable Feeds
Site Standard Pages	Logs when any standard page is enabled or disabled.
Site Error Pages	Logs when any error page assignment is changed.
Login Settings Changes	Changes to the following login settings are logged: • Portal • Change Password Page • Require Non-Secure Connections (HTTP)

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:
- "Customize Application"

Event	Description
URL Redirect Changes	Logs when any URL redirect is created, deleted, enabled, disabled, or changed.

SEE ALSO:

Force.com Sites Overview

Viewing 24-Hour Force.com Sites Usage History

Monitor the bandwidth and request time usage for each of your sites by viewing the usage data tracked on this related list. By closely monitoring your sites, you can avoid exceeding your limits.

To view the 24-hour usage history for your site:

1. From Setup, enter *Sites* in the Quick Find box, then select **Sites**.
2. Click the name of the site you want to view.
3. View the 24-Hour Usage History related list. Usage information may be delayed by up to several minutes due to processing time.

The 24-Hour Usage History related list tracks and displays the following usage metrics for your site:

Metric	How It's Calculated
Origin Bandwidth	Bandwidth is calculated as the number of megabytes served and received from the site's origin server. The Daily Limit applies to a rolling 24-hour period.
Request Time	"Service request time" is calculated as the total server time in minutes required to generate pages for the site. The Daily Limit applies to a rolling 24-hour period.

"Origin server" refers to the Web server that hosts your site. "Rolling 24-hour period" refers to the 24 hours immediately preceding the current time.

For each metric, the related list displays Current Usage, Daily Limit, and the Percent Used.

SEE ALSO:

Tracking Your Force.com Site with Google Analytics

Reporting on Force.com Sites

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

Reporting on Force.com Sites

To keep track of your site activity and usage, take advantage of the Sites Usage Reporting managed package to analyze your monthly page views, daily bandwidth, and daily service request time so you can avoid reaching monthly and daily limits for individual sites, as well as for your organization.

To get started using Salesforce Reports and Dashboards for sites:

1. Install the Sites Usage Reporting managed package.
2. Use packaged reports to analyze site usage.
3. Optionally, create custom reports to analyze site usage.
4. Use the Site Usage Dashboard to monitor sites.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To install packages:
- "Download AppExchange Packages"

To run reports:
- "Run Reports"

  AND

  "Read" on the records included in reports

To create, edit, save, and delete reports:
- "Run Reports" and "Read" on the records included in the reports

  AND

  "Create and Customize Reports"

To create, edit, and delete dashboards:
- "Run Reports"

  AND

  "Manage Dashboards"

Installing the Sites Usage Reporting Managed Package

The Sites Usage Reporting managed package, available on AppExchange, contains out-of-the-box reports and a dashboard for monitoring sites usage.

To find the Sites Usage Reporting managed package, go to AppExchange and search on "sites reporting," or go to `http://sites.force.com/appexchange/listingDetail?listingId=a0N30000001SUEwEAO`.

Using Packaged Reports to Analyze Site Usage

The Sites Usage Reporting managed package contains the following reports for the sites in your organization. You can find these reports in the Site Usage Reports folder under All Reports in the Reports tab. You can also select **Site Usage Reports** in the `Folder` drop-down list, then click **Go**.

Note: Site usage data is aggregated at midnight, GMT, so the current day's page view counts may not be accurately reflected in reports, depending on your time zone. Cache server page views may take a day or more to be reflected in reports.

Report	Description
Current Period Page Views	Shows the total page views for the current period (calendar month), measured against page views allowed. Page views are broken down by site and by day. The current period limit applies to all sites within the organization.
Daily Total Bandwidth Usage	Shows the total bandwidth usage over the last 30 days, broken down by site, by day, and by origin and cache servers.
Daily Total Page Views	Shows the total page views over the last 30 days, broken down site, by day, and by origin and cache servers.
Site Daily Origin Bandwidth Usage	Shows the total origin bandwidth usage over the last 30 days, broken down by site and by day.
Site Daily Request Time Usage	Shows the total origin service request time over the last 30 days, broken down by site and by day.
Top Bandwidth Consuming Sites	Shows the sites that consumed the most bandwidth during the current period.
Top Resource Consuming Sites	Shows the sites that consumed the most service request time during the current period.
Top Sites by Page Views	Shows the sites that generated the most page views during the current period.

Creating Custom Reports to Analyze Site Usage

You can also create custom reports on sites:

1. From the Reports tab, click **New Report**.
2. For the report type, select **Administrative Reports**, then **Site Usage Reports**. You must enable sites for your organization and install the Sites Usage Reporting managed package to see the Site Usage Reports custom report type.
3. Click **Create** to create a custom report. Fields related to your sites, such as `Site Name`, `Site Status`, `Daily Bandwidth Limit`, and `Daily Request Time Limit` can all be used in your custom report.

Note: When you create your own custom reports using the Site Usage Reports custom report type, be aware that the `Origin Bandwidth` column is measured in bytes, and the `Request Time` column is measured in milliseconds. Make sure you consider the difference in units when comparing these columns to the `Daily Bandwidth Limit` and `Daily Request Time Limit` columns, which are measured in megabytes and minutes, respectively.

For the out-of-the-box reports included with the managed package, bandwidth is measured in megabytes and request time is measured in minutes.

Using the Site Usage Dashboard to Monitor Sites

The Sites Usage Reporting managed package contains the Site Usage Dashboard to help you monitor the sites in your organization at a glance. The dashboard contains a component for each of the reports provided in the managed package.

To access the dashboard, from the Dashboards tab:

- Use the `View Dashboard` field.
- Or, click **Go to Dashboard List** and select **Site Usage Dashboard** from the dashboard list.

To modify the dashboard, click **Edit**. You can also create your own custom dashboard using any custom reports you may have created. Consider adding the Site Usage Dashboard as the dashboard snapshot on your home page.

SEE ALSO:

Viewing 24-Hour Force.com Sites Usage History

Tracking Your Force.com Site with Google Analytics

Force.com Sites Security

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Consider the following security issues when setting up your Force.com site:

- If the `Require Secure Connections (HTTPS)` checkbox on the Session Settings page is selected, and the `Require Non-Secure Connections (HTTP)` checkbox on the Login Settings page is not selected, users logging in to the associated portal from the site will see the `secure.force.com` domain. For example, if you registered `mycompany.force.com` as your custom domain, the URL changes to `https://mycompany.secure.force.com` upon login. For more information, see Managing Force.com Site Login and Registration Settings.
- Customers using a script to login to sites can use the optional `refURL` URL parameter to retain the custom domain name after login. This parameter has no effect if `Require Non-Secure Connections (HTTP)` has been set for the site or `Require Secure Connections (HTTPS)` has been set for the organization. An example URL using `refURL` is: `http://mysite.secure.force.com/SiteLogin?refURL=http://mysite.com`.
- To set restrictions based on IP or login hours, HTTPS is required. You must use the secure URL associated with your Force.com domain to access your site.
- To enforce HTTPS on all Force.com sites pages and allow all IP addresses to access your site, create the following IP ranges: `0.0.0.0` to `255.255.255.255`, `::` to `::fffe:ffff:ffff`, and `::1:0:0:0` to `ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`. However, as this may degrade the performance of your site, don't enforce HTTPS unless it is absolutely required. Changing from HTTP to HTTPS doesn't affect logged in users until the next time they log in.
- Authenticated and non-authenticated users may see different error messages for certain conditions—for example, on Apex exceptions.
- Only production organizations have the valid `secure.force.com` SSL certificates to access sites using HTTPS.

  Note: If a site within a sandbox (non-production) organization is accessed using HTTPS, a certificate name mismatch warning may appear.

- Cache settings on static resources are set to private when accessed via a Force.com site whose guest user's profile has restrictions based on IP range or login hours. Sites with guest user profile restrictions cache static resources only within the browser. Also, if a previously unrestricted site becomes restricted, it can take up to 45 days for the static resources to expire from the Salesforce cache and any intermediate caches.

Warning:

- We recommend setting the sharing to private for the objects on which you grant "Read" access for your site. This ensures that users accessing your site can view and edit only the data related to your site.
- We also recommend securing the visibility of all list views. Set the visibility of your list views to `Visible to certain groups of users`, and specify the groups to share to. List views whose visibility is set to `Visible to all users` may be visible to public users of your site. To share a list view with public users, create a new public group for those users and give them visibility. If the object's sharing is set to private, public users won't be able to see those records, regardless of list view visibility.

Force.com Sites-Related Apex Methods and Expressions

Force.com Apex Methods

Apex methods for Force.com Sites are contained in the `site` class, `cookie` class, and `urlRewriter` class. See the *Force.com Apex Code Developer's Guide*.

Force.com Sites-Related Expressions

Force.com also provides the following expressions to use on Visualforce pages, email templates, and s-controls:

Merge Field	Description
$Site.Name	Returns the API name of the current site.
$Site.Domain	Returns the Force.com domain name for your organization.
$Site.CustomWebAddress	Returns the request's custom URL if it doesn't end in `force.com` or returns the site's primary custom URL. If neither exist, then this returns an empty string. Note that the URL's path is always the root, even if the request's custom URL has a path prefix. If the current request is not a site request, then this field returns an empty string. This field's value always ends with a `/` character. Use of $Site.CustomWebAddress is discouraged and we recommend using $Site.BaseCustomUrl instead.
$Site.OriginalUrl	Returns the original URL for this page if it's a designated error page for the site; otherwise, returns `null`.
$Site.CurrentSiteUrl	Returns the base URL of the current site that references and links should use. Note that this field might return the referring page's URL instead of the current request's URL. This field's value includes a path prefix and always ends with a `/` character. If the current request is not a site request, then this field returns an empty string. Use of $Site.CurrentSiteUrl is discouraged. Use $Site.BaseUrl instead.
$Site.LoginEnabled	Returns `true` if the current site is associated with an active login-enabled portal; otherwise returns `false`.
$Site.RegistrationEnabled	Returns `true` if the current site is associated with an active self-registration-enabled Customer Portal; otherwise returns `false`.
$Site.IsPasswordExpired	For authenticated users, returns `true` if the currently logged-in user's password is expired. For non-authenticated users, returns `false`.
$Site.AdminEmailAddress	Returns the value of the `Site Contact` field for the current site.

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit Force.com sites:

- "Customize Application"

Merge Field	Description
$Site.Prefix	Returns the URL path prefix of the current site. For example, if your site URL is `myco.force.com/partners`, `/partners` is the path prefix. Returns `null` if the prefix isn't defined. If the current request is not a site request, then this field returns an empty string.
$Site.Template	Returns the template name associated with the current site; returns the default template if no template has been designated.
$Site.ErrorMessage	Returns an error message for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.
$Site.ErrorDescription	Returns the error description for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.
$Site.AnalyticsTrackingCode	The tracking code associated with your site. This code can be used by services like Google Analytics to track page request data for your site.
$Site.BaseCustomUrl	Returns a base URL for the current site that doesn't use a Force.com subdomain. The returned URL uses the same protocol (HTTP or HTTPS) as the current request if at least one non-Force.com custom URL that supports HTTPS exists on the site. The returned value never ends with a `/` character. If all the custom URLs in this site end in `force.com`, or this site has no custom URL's, then this returns an empty string. If the current request is not a site request, then this method returns an empty string. This field replaces `CustomWebAddress` and includes the custom URL's path prefix.
$Site.BaseInsecureUrl	Returns a base URL for the current site that uses HTTP instead of HTTPS. The current request's domain is used. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.
$Site.BaseRequestUrl	Returns the base URL of the current site for the requested URL. This isn't influenced by the referring page's URL. The returned URL uses the same protocol (HTTP or HTTPS) as the current request. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.
$Site.BaseSecureUrl	Returns a base URL for the current site that uses HTTPS instead of HTTP. The current request's domain is preferred if it supports HTTPS. Domains that are not Force.com subdomains are preferred over Force.com subdomains. A Force.com subdomain, if associated with the site, is used if no other HTTPS domains exist in the current site. If there are no HTTPS custom URLs in the site, then this method returns an empty string. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.
$Site.BaseUrl	Returns the base URL of the current site that references and links should use. Note that this field may return the referring page's URL instead of the current request's URL. This field's value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this field returns an empty string.

Merge Field	Description
	This field replaces $Site.CurrentSiteUrl.
$Site.MasterLabel	Returns the value of the Master Label field for the current site. If the current request is not a site request, then this field returns an empty string.
$Site.SiteId	Returns the ID of the current site. If the current request is not a site request, then this field returns an empty string.
$Site.SiteType	Returns the API value of the Site Type field for the current site. If the current request is not a site request, then this field returns an empty string.
$Site.SiteTypeLabel	Returns the value of the Site Type field's label for the current site. If the current request is not a site request, then this field returns an empty string.

Note: To use these expressions, the Force.com sites feature must be enabled for your organization. You must also use them within the context of your public site; otherwise, an empty string is returned for all expressions except {!$Site.Template}, which returns the default template for the site.

SEE ALSO:

Configuring Force.com Sites

Force.com Sites Considerations

Before creating and using Force.com sites, consider the following:

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Packaging

The following apply to packaging for sites:

- Sites are not packageable. However, you can package sample code, Visualforce pages, Apex classes, or components for a site using a managed package.
- You must have Force.com sites enabled to be able to install unmanaged packages that contain Visualforce pages or Apex classes that refer to a site.

Access and Authentication

You can grant "Read" and "Create" permissions on all standard objects except products, price books, and ideas; and "Read," "Create," "Edit," and "Delete" on all custom objects. For additional access, you must authenticate site visitors as portal users.

Custom authentication is not supported. You can authenticate via the following:

- Customer Portals—enable public login and registration
- Partner portals—create partner users

Tip: You can also enable single sign-on for portals, as well as Sites.

SEE ALSO:

Setting Up Force.com Sites

Force.com Sites Limits and Billing

Force.com sites usage is governed by monthly and daily limits. Understanding these limits is important to the success of your sites. Salesforce provides tools to help you reduce bandwidth consumption and monitor site usage so that you can avoid exceeding these limits.

This topic contains the following sections:

- Limits and Billing Terminology
- Sites Limits and Restrictions
- Bandwidth and Service Request Time Limit Enforcement
- Billing and Monthly Page Views Enforcement
- What Counts as a Page View?
- Monitoring Limits

EDITIONS

Available in: Salesforce Classic

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Limits and Billing Terminology

This section defines the terminology used for Force.com sites limits and billing:

- "Page Views" are calculated as the total number of pages served from either the site's origin server or the cache server.
- "Bandwidth" is calculated as the number of megabytes served and received from both the site's origin server and the cache server.
- "Service request time" is calculated as the total server time in minutes required to generate pages for the site.
- "Rolling 24-hour period" refers to the 24 hours immediately preceding the current time.
- "Origin server" refers to the Web server that hosts your site.
- "Cache server" refers to the CDN server that serves your cached site pages.
- "Current period" refers to the current calendar month for which you are entitled a certain number of page views for your organization.

Sites Limits and Restrictions

The following table lists the site limits for each edition:

Edition	Maximum Number of Sites	Bandwidth Limit (per rolling 24-hour period per site)	Service Request Time (per rolling 24-hour period per site)	Maximum Page Views
Developer Edition	1	500 MB	10 minutes	N/A
Enterprise Edition	25	1 GB for sandbox 40 GB for production	30 minutes for sandbox 60 hours for production	500,000
Unlimited Edition Performance Edition	25	1 GB for sandbox 40 GB for production	30 minutes for sandbox 60 hours for production	1,000,000

Make sure to consider all of the available caching options to help you avoid reaching these limits, and use the Site Usage analytics tools to monitor your sites.

Bandwidth and Service Request Time Limit Enforcement

Bandwidth and Service Request Time limits are tracked and enforced over a 24-hour period. Sites that exceed provisioned limits within the 24-hour period remain available if the host instance has resources to serve the site. However, even if a site is available once limits are exceeded, there's no guarantee in service level.

Billing and Monthly Page Views Enforcement

This section describes how Salesforce enforces limits on monthly page views for sites:

- Billing is based on the number of monthly page views purchased for your organization. This page view limit is cumulative for all sites in your organization.
- If, in a given calendar month, your organization reaches 110% of its page view limit, Salesforce sends the site and billing administrators email notification.
- If your organization exceeds 110% of its page view limit for four consecutive calendar months, your sites are disabled until the next calendar month begins or you purchase more page views. Also, Salesforce sends email notification to the site and billing administrators, as well as the related account executive.
- If, in a given calendar month, your organization reaches 300% of its page view limit, your sites are disabled until the next calendar month begins or you purchase more page views. Also, Salesforce sends email notification to the site and billing administrators, as well as the related account executive.

What Counts as a Page View?

This section describes how page views are counted for sites.

A page view is a request from a non-authenticated site user to load a page associated with one of the sites within your Force.com domain or custom domain. Requests from authenticated portal users are not counted as page views.

Requests that Count as Page Views

Requests for the following *are* counted as page views:

Requests for...	Example URL
Your Force.com domain	`http://mycompany.force.com`
Your custom Web address	`http://mycompany.com`
Any page associated with your site	`http://mycompany.force.com/mypage`
Authorization Required error page	`http://mycompany.force.com/Unauthorized`
AJAX requests such as: • JavaScript remoting (for example, Apex `RemoteAction`) • Visualforce `<apex:actionFunction>` • Visualforce `<apex:actionPoller>`	n/a

Requests that Do Not Count as Page Views

Requests for the following *are not* counted as page views:

Requests for...	Example URL
Salesforce images	`http://mycompany.force.com/img/force_logo_w09.gif`
Your static resources	`http://mycompany.force.com/resource/1233771498000/background`
Robots.txt	`http://mycompany.force.com/robots.txt`
Favorite icon	`http://mycompany.force.com/favicon.ico`
Attachments and Documents	n/a
Error pages, apart from Authorization Required, such as Limit Exceeded and Maintenance	`http://mycompany.force.com/BandwidthExceeded`
Images included with an HTML field	`http://mycompany.force.com/servlet/rtaImage`
Custom file field	`http://mycompany.force.com/servlet/fileField`

Note: Limit Exceeded, Maintenance, Page Not Found, and designated Inactive Home pages aren't counted against page view and bandwidth limits. You can use static resources to brand these pages, but the following limitations apply:

- Static resources must be 50 KB or smaller in size.
- Static resources must be style sheets (CSS), image files, or JavaScript files.
- You can't use Apex controllers for these pages.
- You can't perform SOQL or DML operations from these pages.

Monitoring Limits

Page views, bandwidth, and time consumption are tracked and made available in your org. You can view this information for a site under **Setup** > **Build** > **Develop** > **Sites**. Select a site, and you'll see related lists for page views for the current month's billing cycle, and the 24-hour bandwidth and service request time usage history.

Additionally, you can install the Force.com Sites Usage Reporting app from AppExchange to monitor usage. Keep in mind that the information available in the app might not be as current as the information you'll find directly in your org.

For more information about bandwidth and service request time, see Viewing 24-Hour Force.com Sites Usage History.

SEE ALSO:

Viewing 24-Hour Force.com Sites Usage History

Tracking Your Force.com Site with Google Analytics

Can I use the same domain name for my Force.com Sites and my Communities?

No, you can't use the same domain name.

Force.com Sites and Communities must each use a unique domain name. If you're using a domain name for your Force.com site and you want to use it for your communities instead, contact Salesforce Support for assistance with renaming the domain.

Extending the Reach of Your Organization

Actions, Buttons, and Links

Buttons and links let users interact with Salesforce data and with external websites and services, such as search engines and online maps. Salesforce includes several standard buttons and links. You can also create custom ones. Actions let users do tasks, such as create records in the Chatter publisher and in Salesforce1.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

USER PERMISSIONS

To create or edit buttons, links, and actions:
- "Customize Application"

IN THIS SECTION:

Actions

Actions add functionality to Salesforce. Choose from standard actions, such as create and update actions, or create actions based on your company's needs.

Custom Buttons and Links

Custom buttons and links help you integrate Salesforce data with external URLs, applications, your company's intranet, or other back-end office systems.

Actions

Actions add functionality to Salesforce. Choose from standard actions, such as create and update actions, or create actions based on your company's needs.

Create actions and add them to the Chatter publisher on the home page, on the Chatter tab, in Chatter groups, and on record detail pages. In Salesforce Classic, actions appear in the Chatter publisher. In Lightning Experience, actions appear in different areas of the user interface, depending on the action's type. In Salesforce1, actions appear in the action bar, its associated action menu, and as list-item actions.

Actions enable users to do more in Salesforce and in Salesforce1. For example, create or update records and log calls directly in the Chatter feed or from users' mobile devices.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Action Types

There are several categories of actions, such as standard Chatter actions, nonstandard actions, default actions, mobile smart actions, custom actions, and productivity actions.

Enable Actions in the Chatter Publisher

Enabling actions in the publisher lets you add actions you've created to Chatter publishers on the Home page, on the Chatter tab, in Chatter groups, and on record detail pages.

Action Layout Editor

Just as object record pages have page layouts that can be customized, actions have action layouts that can be customized. When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they're taking the action.

Custom Success Messages for Quick Actions

For Create a Record, Update a Record, and Log a Call action types, you can create a custom message that displays when the action executes successfully.

Customize Actions with the Enhanced Page Layout Editor

Use the page layout editor to customize which actions show up in Salesforce and in Salesforce1.

Set Predefined Field Values for Quick Action Fields

When you create actions, use predefined field values to set a value for a field. Predefined values can help ensure consistency and make it faster and easier for users to create records.

Global Publisher Layouts

Actions in Lightning Experience

In Lightning Experience, actions display in the Global Actions menu in the header, on list view items, and in several places on a record page. Where they display on a record page depends on the action's type.

Salesforce1 Action Bar

Salesforce1 users have a one-stop place to find actions, so there's no confusion about where to go to do something. The action bar and its associated action menu collect actions from different places in Salesforce1 into a single, unified home.

Quick Actions and Record Types

Using record types in your organization can affect the availability of quick actions for your users.

Actions Best Practices

Troubleshooting Actions

SEE ALSO:

Set Up Actions with Chatter Enabled

Actions Best Practices

Action Types

There are several categories of actions, such as standard Chatter actions, nonstandard actions, default actions, mobile smart actions, custom actions, and productivity actions.

Which actions are available in the full Salesforce site depends on whether your organization has Chatter, feed tracking, and actions in the publisher enabled. Actions in Salesforce1 don't rely on whether Chatter or actions in the publisher are enabled. For how Chatter enablement affects action visibility, see Actions with and without Chatter on page 876.

Category	Description	Included Actions	Where They Display
Standard Chatter actions	Standard Chatter actions are included when Chatter is enabled. You can customize the order in which these actions appear, but you can't edit their properties. Standard Chatter actions require that feed tracking for objects is enabled.	Post, File, Link, Poll, Question, Thanks (Work.com), and Announcements (Groups)	Salesforce and Salesforce1 Only Post and Announcements are supported in Lightning Experience

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Category	Description	Included Actions	Where They Display
Nonstandard actions	Nonstandard actions are actions that you create and customize yourself. The actions can be global or object-specific.		Salesforce and Salesforce1
Default actions	Default actions are Salesforce predefined actions to get you and your users started using actions in your organization. Add default actions to publisher layouts to make them available to your users in the full Salesforce site and the action bar in Salesforce1. Default actions are supported on account, case, contact, lead, and opportunity objects.	Depends on the object. For a list of global actions and which default actions apply to which object, see Default Actions on page 857.	Salesforce and Salesforce1
Mobile smart actions	Mobile smart actions are a set of preconfigured actions, just like default actions, and are supported on the same list of objects. Mobile smart actions appear as a single element in the page layout editor. In Salesforce1, the Mobile Smart Actions element expands to a set of distinct create actions that enable users to create records directly in the feed.	Depends on the object. For a list of which actions the Mobile Smart Actions element expands to include for each object, see Mobile Smart Actions on page 860.	Salesforce1
Custom actions	*Custom actions* invoke Lightning components, Visualforce pages, or canvas apps with functionality that you define. For example, you can create a custom action so that users can write comments that are longer than 5,000 characters, or create one that integrates a video-conferencing application so that support agents can communicate visually with customers. Custom actions can be global or object-specific.	Custom actions that you create.	Salesforce and Salesforce1 Not supported in Chatter groups with customers
Productivity actions	Productivity actions are predefined by Salesforce and are attached to a limited set of objects. You can't edit or delete productivity actions. Productivity actions appear on these objects. • Account • Contact • Event • Lead	Depends on the object. The actions include Send Email, Log a Call, Map, View Website, and Read News.	Salesforce (Lightning Experience only) and Salesforce1

Category	Description	Included Actions	Where They Display
	• User • User Profile		

Within the categories of actions, you can have different types of actions, depending on their function.

- *Create actions* let users create records. They're different from the Quick Create and Create New features on the Salesforce home page, because create actions respect validation rules and field requiredness, and you can choose each action's fields.
- *Log a call actions* let users record the details of phone calls or other customer interactions. These call logs are saved as completed tasks.
- *Question actions* enable users to ask and search for questions about the records that they're working with.
- *Send email actions*, available only on cases, give users access to a simplified version of the Case Feed Email action on Salesforce1.
- *Update actions* let users make changes to a record.

For create, log-a-call, and custom actions, you can create either object-specific actions or global actions. Update actions must be object-specific.

IN THIS SECTION:

Default Actions

Default actions are Salesforce predefined actions to get you and your users started using actions in your organization. Add default actions to publisher layouts to make them available to your users in the full Salesforce site and the action bar in Salesforce1.

Mobile Smart Actions

Mobile smart actions are quick actions that are available for account, case, contact, lead, and opportunity pages and on the global publisher layout in Salesforce1. You can use them to set up quick actions for mobile users with little effort.

Global Actions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab, and object pages. Global create actions enable users to create object records, but the new record has no relationship with other records.

Object-Specific Actions

Object-specific actions let users quickly create or update records, log calls, send emails, and more, in the context of a particular object.

Custom Quick Actions

Custom actions invoke Lightning components, Visualforce pages, or canvas apps with functionality that you define. For example, you can create a custom action so that users can write comments that are longer than 5,000 characters, or create one that integrates a video-conferencing application so that support agents can communicate visually with customers.

Default Actions

Default actions are Salesforce predefined actions to get you and your users started using actions in your organization. Add default actions to publisher layouts to make them available to your users in the full Salesforce site and the action bar in Salesforce1.

Each default action has a predefined set of fields. Use the page layout editor or global publisher layout to remove actions or to change the order in which the actions appear.

This table lists the available default actions. Italicized actions are standard Chatter actions.

Note: In organizations that are created after Winter '14, Salesforce adds default actions to the global publisher layout and to the publisher layout on account, case, contact, lead, and opportunity objects. In organizations that were created before Winter '14, default actions are available in the palette on the page layout editor, but they're not automatically added to publisher layouts.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, and **Developer** Editions

Layout Page	Default Actions
Global layout (also applies to custom objects)	• *Post* • *File* • New Event • New Task • New Contact • Log a Call (logged calls are saved as completed tasks) • New Opportunity • New Case • New Lead • *Thanks* • *Link* • *Poll* • *Question*
Account	• *Post* • *File* • New Event • New Task • New Contact • New Case • Log a Call (logged calls are saved as completed tasks) • New Note • New Opportunity • *Thanks* • *Link* • *Poll* • *Question*

Layout Page	Default Actions
Case	• *Post* • *File* • New Child Case • Log a Call (logged calls are saved as completed tasks) • New Task • New Event • *Thanks* • *Link* • *Poll* • *Question*
Contact, Lead, and Opportunity	• *Post* • *File* • New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Even • *Thanks* • *Link* • *Poll* • *Question*

Note: Using record types in your organization can affect the availability of global default actions for your users. For more information, see Quick Actions and Record Types on page 902.

IN THIS SECTION:

Default Action Fields

Each default action includes a pre-defined set of fields, which lets you make actions available to your users without spending a lot of time on setup.

SEE ALSO:

Actions

Default Action Fields

Each default action includes a pre-defined set of fields, which lets you make actions available to your users without spending a lot of time on setup.

These fields are included on each default action and appear in the action layout in the order listed.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, and **Developer** Editions

Action	Fields
Log a Call	• Subject (default value: Call) • Comment (description) • Name • Related To
New Case and New Child Case	• Contact Name • Status • Subject • Description
New Contact	• Name • Email • Phone • Account Name • Title
New Event	• Subject • Start • End • All-Day Event • Name • Related To • Assigned To • Location
New Lead	• Name • Email • Phone • Company • Title
New Note	• Title • Public (checkbox) • Body

Action	Fields
New Opportunity	• Opportunity Name • Account • Close Date (default value: 30 days from today) • Stage • Amount
New Task	• Subject • Due Date • Name • Related To • Assigned To • Status

You can change the fields that appear on each action layout using the action layout editor.

SEE ALSO:

Default Actions

Actions

Mobile Smart Actions

Mobile smart actions are quick actions that are available for account, case, contact, lead, and opportunity pages and on the global publisher layout in Salesforce1. You can use them to set up quick actions for mobile users with little effort.

Note: Mobile smart actions don't appear in the full Salesforce site, regardless of which page layouts you add them to. They appear only to users in Salesforce1. If you have users who access Salesforce on mobile devices more frequently than they access the full site, consider creating specific page layouts for them and including mobile smart actions only on those layouts.

Mobile smart actions are populated with all your organization's required fields on the relevant object, regardless of how many fields there are. For example, the New Case action in the mobile smart action bundle includes all required case fields. You can't edit the fields on mobile smart actions. The fields that appear change only if you change which fields on an object are required.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

You also can't change which actions are included as part of a mobile smart actions bundle—removing New Event or adding a custom action, for example. To create a more customized set of actions, create the actions you want, add them to the relevant page layouts, and remove the mobile smart actions bundle.

Mobile smart actions appear as a single action element in the page layout editor. However, they appear in the action bar and action menu in Salesforce1 as distinct create actions. These distinct actions allow users to create records directly from the action bar.

Here's what the mobile smart action element on each supported object expands to include. (The actions appear in the action bar and menu in the order shown.)

Layout	Actions Included in Mobile Smart Action Bundle
Global layout (also applies to custom objects)	• New Task • New Contact • Log a Call (logged calls are saved as completed tasks) • New Opportunity • New Case • New Lead
Account	• New Task • New Contact • New Case • Log a Call (logged calls are saved as completed tasks) • New Note • New Opportunity
Case	• New Child Case • Log a Call (logged calls are saved as completed tasks) • New Task • New Event
Contact	• New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event
Lead	• New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event
Opportunity	• New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note

Layout	Actions Included in Mobile Smart Action Bundle
	• New Event

SEE ALSO:

Actions

Set Up Actions with Chatter Enabled

Global Actions

EDITIONS

Available in: Salesforce Classic

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab, and object pages. Global create actions enable users to create object records, but the new record has no relationship with other records.

You can also use a global `Create a Record` action to enable your Salesforce for Outlook users to create records directly from the Salesforce side panel. For more information, see Set Up Reps to Create Records Directly from the Salesforce Side Panel.

Note: Only standard Chatter actions (Post, File, Link, Poll, and Thanks) appear on the user profile page, regardless of the actions in the User page layout.

To let users ask questions in their Chatter feed, groups, and records, add the Question action to global layouts.

Add Log a Call actions to global layouts to let users record call details from global pages. For example, users can log calls from the Home page and the Chatter tab in the full Salesforce site, or the Feed or Groups pages in Salesforce1.

Use a Visualforce page, Lightning component, or a canvas app to create global custom actions for tasks that don't require users to use records that have a relationship to a specific object. Canvas apps that you want to use as custom actions require Publisher as a location. To call Lightning components from custom actions, you must have My Domain deployed in your org. Visualforce pages that you want to use as global custom actions can't use standard controllers. For example, you want a custom action that lets users enter a street address and see a map, the local time, and the local weather. For this action, create a Visualforce page that doesn't use any of the standard controllers, and add it as a custom global action.

You can find out more about canvas apps and custom actions in the *Salesforce1 App Developer Guide*.

Note:

- Chatter groups with customers don't support global create, log a call, or custom actions and display only standard Chatter actions, such as Post, File, Link, and Poll.
- Actions to create records for an object that is the detail object in a master-detail relationship must be object-specific, not global.

Supported Objects for Create Actions

You can create global actions to let users create many kinds of records, including:

- Account
- Asset
- Badge
- Campaign
- Case

- Contact
- Contract
- Custom objects
- Event (without invitees)
- Goal
- Group
- Lead
- Note
- Opportunity
- Person Account
- Question
- Reward
- Task
- Work Order

IN THIS SECTION:

Create Global Quick Actions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab, and object pages. For example, add a Log a Call action to global layouts to let users record call details right from a Chatter thread.

SEE ALSO:

Object-Specific Actions

Actions with and without Chatter

Actions

Create Global Quick Actions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab, and object pages. For example, add a Log a Call action to global layouts to let users record call details right from a Chatter thread.

USER PERMISSIONS

To create actions:
- "Customize Application"

Walk Through It: Create a Global Quick Action

1. From Setup, enter *Actions* in the Quick Find box, then select **Global Actions**.
2. Click **New Action**.
3. Select the type of action to create.
4. Customize the action.
   - For a Create a Record action, select the type of object to create. If the object has more than one record type, select the one you want to use for records created through this action.
   - For a Custom Visualforce or Custom Canvas action, select a Visualforce page or canvas app, and then specify the height of the action window. The width is fixed.
   - For a Lightning Component action, select the component to be called by the action.

5. Enter a label for the action. Users see this label as the name of the action.

   Tip: If you're creating a Create a Record or Log a Call action, choose an option from the `Standard Label Type` list to have Salesforce generate the label. For the labels in this list that include "Record" and "Record Type", Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New "Record" standard label on a create contact action, the generated label is Create New Contact.

6. If necessary, change the name of the action.
   This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we suggest not editing this field.
7. Type a description for the action.
   The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, we recommend using a detailed description, such as "Create Contact on Account using New Client record type."
8. For a Create a Record or Log a Call action, select whether you want a feed item to be created when the action is completed.
9. For a Create a Record, Update a Record, or Log a Call action, you can add a custom success message that displays after the action executes successfully.
10. Optionally, click **Change Icon** to select a different icon for the action.
    Custom images used for action icons must be less than 1 MB in size.
11. Click **Save**.

After you create a quick action, customize its layout, add predefined values, and then add the action to page layouts.

Tip: If you delete an action, the action is removed from all layouts that it's assigned to.

SEE ALSO:

Set Predefined Field Values for Quick Action Fields

Customize Actions with the Enhanced Page Layout Editor

Custom Success Messages for Quick Actions

Visualforce Pages as Object-Specific Custom Actions

Object-Specific Actions

Object-specific actions let users quickly create or update records, log calls, send emails, and more, in the context of a particular object.

Object-specific create actions create records that are automatically associated with related records. For example, you add an object-specific action on the Account object that creates contacts. If a user creates a contact with that action on the detail page for the Acme account, that new contact is automatically associated with Acme. Object-specific actions are only available on page layouts for that object. For example, you can add the New Group Member action only to the group publisher layout.

When a user creates a record by using an object-specific create action, a feed item for that record appears:

- In the feed for the record on which the new record was created
- As the first entry in the feed for the new record
- In the Chatter feed of the user who created the record
- In the user profile feed for the user who created the record
- In the Chatter feed of any users who follow the record on which the record was created
- In the Chatter feed of any users who, through custom triggers or auto-follow rules for new records, automatically follow the new record

Object-specific Log a Call actions let users enter notes about calls, meetings, or other interactions that are related to a specific record.

Object-specific update actions make it easy for users to edit records. You can define the fields that are available for update.

Object-specific custom actions invoke Lightning components, Visualforce pages, or canvas apps that let users interact with or create records that have a relationship to an object record. The Visualforce page for an object-specific custom action must include the standard controller for the relevant object. For example, use the standard contact controller to create a custom action that lets users import a contact's Twitter profile and add that information to a contact record.

Send email actions, available only on cases, give users access to a simplified version of the Case Feed Email action on Salesforce1.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Supported Objects

When you create an object-specific action, you can choose as a target object only an event, task, or any object that has a parent-child or lookup relationship to the host object. You can create object-specific actions on many objects, including:

- Account
- Campaign
- Case
- Contact
- Custom objects
- Group
- Lead
- Opportunity

IN THIS SECTION:

Create Object-Specific Quick Actions

Object-specific actions let users create records that have automatic relationships to other records, make updates to specific records, and interact with records in ways that you define.

SEE ALSO:

Global Actions

Actions with and without Chatter

Actions

Create Object-Specific Quick Actions

Object-specific actions let users create records that have automatic relationships to other records, make updates to specific records, and interact with records in ways that you define.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To create actions:
- "Customize Application"

1. From the management settings for the object for which you want to create an action, go to Buttons, Links, and Actions.
2. Click **New Action**.
3. Select the type of action to create.
4. Customize the action.
   - For a Create a Record action, select the type of object to create.
     - If the object has more than one record type, select the one you want to use for records created through this action.
     - If the object for which you're creating the action has more than one relationship with the target object, select the field you want to populate when a record is created. If the two objects have a master-detail relationship, you can't select which field to populate. The master-detail `Relationship` field is selected by default, and you can't change this setting.
   - For a Custom Visualforce action, select the Visualforce page, and then specify the height of the action window. The width is fixed.
   - For a Lightning Component action, select the component called by the action.
5. Enter a label for the action. Users see this label as the name of the action.

   Tip: If you're creating a Create a Record or Log a Call action, choose an option from the `Standard Label Type` list to have Salesforce generate the label. For the labels in this list that include "Record" and "Record Type", Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New "Record" standard label on a create contact action, the generated label is Create New Contact.
6. If necessary, change the name of the action.
   This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we suggest not editing this field.
7. Type a description for the action.

The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, we recommend using a detailed description, such as "Create Contact on Account using New Client record type."

8. For a Create a Record or Log a Call action, select whether you want a feed item to be created when the action is completed.
9. For a Create a Record, Update a Record, or Log a Call action, you can add a custom success message that displays after the action executes successfully.
10. Optionally, click **Change Icon** to select a different icon for the action.

    Custom images used for action icons must be less than 1 MB in size.
11. Click **Save**.

After you create a quick action, customize its layout, add predefined values, and then add the action to page layouts.

Note: If you delete an action, the action is removed from all layouts that it's assigned to.

SEE ALSO:

- Set Predefined Field Values for Quick Action Fields
- Customize Actions with the Enhanced Page Layout Editor
- Visualforce Pages as Object-Specific Custom Actions
- Custom Success Messages for Quick Actions

Custom Quick Actions

Custom actions invoke Lightning components, Visualforce pages, or canvas apps with functionality that you define. For example, you can create a custom action so that users can write comments that are longer than 5,000 characters, or create one that integrates a video-conferencing application so that support agents can communicate visually with customers.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

IN THIS SECTION:

Lightning Component Actions

Lightning component actions are custom actions that invoke a Lightning component. They support Apex and JavaScript and provide a secure way to build client-side custom functionality. Lightning component actions are supported only in Salesforce1 and Lightning Experience.

Visualforce Pages as Object-Specific Custom Actions

Visualforce pages added as custom actions on an object are invoked in the context of a record of that object type. The custom action has a specific record ID handed to it—the record the user was looking at when they clicked the custom action. Design the page to act on that specific record type.

Visualforce Pages as Global Custom Actions

Visualforce pages used as global actions can be invoked in many different places and don't have a specific record associated with them. They have complete "freedom of action," which means it's up to you to write the code.

Hide the Action Header for Visualforce Custom Actions

When creating a Visualforce page to use as a custom action, you can choose to hide the action's header. Hiding the action header helps prevent user confusion, especially if you have your own buttons specified in the Visualforce page.

Prerequisites for Using Canvas Apps as Custom Actions

Using canvas apps as custom actions makes it easy to give users access to the functionality of your apps in Chatter and elsewhere in Salesforce.

SEE ALSO:

Create Global Quick Actions

Create Object-Specific Quick Actions

Visualforce Pages as Object-Specific Custom Actions

Prerequisites for Using Canvas Apps as Custom Actions

Lightning Component Actions

Lightning component actions are custom actions that invoke a Lightning component. They support Apex and JavaScript and provide a secure way to build client-side custom functionality. Lightning component actions are supported only in Salesforce1 and Lightning Experience.

Note: My Domain must be deployed in your org for Lightning component actions to work properly.

You can add Lightning component actions to an object's page layout using the page layout editor. If you have Lightning component actions in your org, you can find them in the Salesforce1 & Lightning Actions category in the page layout editor's palette.

On Lightning Experience record pages, Lightning component actions display in the page-level action menu in the highlights panel.

Lightning component actions can't call just any Lightning component in your org. For a component to work as a Lightning component action, it has to be configured specifically for that purpose and implement either the `force:LightningQuickAction` or `force:LightningQuickActionWithoutHeader` interfaces. You can find out more about configuring custom components in the Lightning Components Developer Guide.

If you plan on packaging a Lightning component action, the component the action invokes must be marked as `access=global`.

EDITIONS

Available in: both Salesforce1 and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, and **Developer** Editions

SEE ALSO:

Actions in Lightning Experience

Visualforce Pages as Object-Specific Custom Actions

Visualforce pages added as custom actions on an object are invoked in the context of a record of that object type. The custom action has a specific record ID handed to it—the record the user was looking at when they clicked the custom action. Design the page to act on that specific record type.

Visualforce pages you create to use as object-specific actions must use a standard object controller. Use controller extensions to add custom code, including `@RemoteAction` methods you can call using JavaScript remoting.

Your custom code could do more than make updates to the originating record. For example, the Create Quick Order custom action searches for matching merchandise. It then creates an invoice and line item, all as part of creating an order for a part. That logic occurs in the context of the originating account record—the invoice is associated to the account record where the quick order action was invoked.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

When the action completes, redirect the user to a page related to the originating record.

The following code sample shows a page designed to be used as a custom action on the account object, so it uses the standard Account controller. This action lets users create cases from account detail pages, and it has a different user interface from standard create actions.

```
public with sharing class CreateCaseExtension {
    private final SObject parent;
    public Case theCase {get; set;}
    public String lastError {get; set;}

    public CreateCaseExtension2(ApexPages.StandardController controller) {
        parent = controller.getRecord();
        theCase = new Case();
        theCase.accountId = parent.id;
        lastError = '';
    }

    public PageReference createCase() {
        createNewCase();
        theCase = new Case();
        theCase.accountId = parent.id;
        return null;
    }

     private void createNewCase() {
        try {
            insert theCase;

            FeedItem post = new FeedItem();
            post.ParentId = ApexPages.currentPage().getParameters().get('id');
            post.Body = 'created a case';
            post.type = 'LinkPost';
            post.LinkUrl = '/' + theCase.id;
            post.Title = theCase.Subject;
            insert post;
        } catch(System.Exception ex){
           lastError = ex.getMessage();
        }
    }
}

<apex:page standardcontroller="Account" extensions="CreateCaseExtension" showHeader="false">

    <script type='text/javascript' src='/canvas/sdk/js/publisher.js'/>
    <style>
        .requiredInput .requiredBlock, .requiredBlock {background-color: white;}
        .custompubblock div {display: inline-block;}
        .custompublabel {width:54px;}
    </style>
    <script>
        function refreshFeed() {
            Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload :
{feed:true}});
        }
    </script>
```

```
    <div>
        <apex:form >
            <apex:actionFunction action="{!createCase}" name="createCase" rerender="out"

            oncomplete="refreshFeed();"/>
            <apex:outputPanel id="out" >
                <div class="custompubblock">
                    <div class="custompublabel">Account:</div><apex:inputField
value="{!theCase.accountId}"
                     style="margin-left:0;"/>   
                   <div>Contact: </div><apex:inputField value="{!theCase.contactId}"
 />
                </div>
                <apex:inputField value="{!theCase.description}"
style="width:538px;height:92px;margin-top:4px;" />
                <div class="custompubblock" style="margin-top:5px;">
                    <div>Status: </div><apex:inputField value="{!theCase.status}"
/>   
                   <div>Priority: </div><apex:inputField value="{!theCase.priority}"
 />   
                  <div>Case Origin: </div><apex:inputField value="{!theCase.origin}"
 />
                </div>
            </apex:outputPanel>
        </apex:form><br/>
        <button type="button" onclick="createCase();"
        style="position:fixed;bottom:0px;right:0px;padding:5px 10px;
        font-size:13px; font-weight:bold; line-height:
        18px;background-color:#0271BF;background-image:-moz-linear-gradient(#2DADDC,
#0271BF);background-repeat:repeat-x;border-color:#096EB3;"
        id="addcasebutton">Create Case</button>
    </div>
</apex:page>
```

Requirements for Refreshing Host Pages

If you want a custom action, whether object-specific or global, to refresh the feed on the page that hosts it, the Visualforce page you create to use as that action must:

- Reference the publisher JavaScript file: `<script type='text/javascript' src='/canvas/sdk/js/publisher.js'/>`. (Creating custom Visualforce actions doesn't require the Force.com Canvas SDK.)
- Include this JavaScript call: `Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed:true}});`.

SEE ALSO:

Visualforce Pages as Global Custom Actions

Create Object-Specific Quick Actions

Actions

Visualforce Pages as Global Custom Actions

Visualforce pages used as global actions can be invoked in many different places and don't have a specific record associated with them. They have complete "freedom of action," which means it's up to you to write the code.

Visualforce pages you create to use as global actions can't use a standard object controller. You must write a custom controller to handle the page.

When a global action completes, the user should be either redirected to a parent record created as part of the action or returned to where they started.

This code sample shows a Visualforce page designed to be used as a custom action on any object that supports actions. This action lets users create cases from record detail pages, Chatter, Chatter groups (except customer groups), or the home page. It has a different user interface from standard create actions. As with all global actions, the records created through this action aren't associated with other records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

```
<!-- Custom controller -->
public with sharing class CreateCaseController {
    public Case theCase {get; set;}
    public String lastError {get; set;}
    public CreateCaseController() {
        theCase = new Case();
        lastError = '';
    }

    public PageReference createCase() {
        createNewCase();
        theCase = new Case();
        return null;
    }

     private void createNewCase() {
        try {
            insert theCase;

            FeedItem post = new FeedItem();
            post.ParentId = ApexPages.currentPage().getParameters().get('id');
            post.Body = 'created a case';
            post.type = 'LinkPost';
            post.LinkUrl = '/' + theCase.id;
            post.Title = theCase.Subject;
            insert post;
        } catch(System.Exception ex){
           lastError = ex.getMessage();
        }
    }
}

<apex:page controller="CreateCaseController" showHeader="false">
    <script type='text/javascript' src='/canvas/sdk/js/publisher.js'/>
    <style>
        .requiredInput .requiredBlock, .requiredBlock {background-color: white;}
        .custompubblock div {display: inline-block;}
        .custompublabel {width:54px;}
```

```
    </style>

     <script>
      function refreshFeed() {
          Sfdc.canvas.publisher.publish({name : 'publisher.refresh',  payload : {feed:
true}});
      }
     </script>


    <div>
        <apex:form >
            <apex:actionFunction action="{!createCase}" name="createCase" rerender="out"
oncomplete="refreshFeed();"/>
            <apex:outputPanel id="out" >
                <div class="custompubblock">
                    <div>Subject: </div><apex:inputField value="{!theCase.subject}"
style="width:500px;" />
                </div>
                <div class="custompubblock">
                    <div class="custompublabel">Account:</div><apex:inputField
value="{!theCase.accountId}"
                    style="margin-left:0;"/>   
                   <div>Contact: </div><apex:inputField value="{!theCase.contactId}"
 />
                </div>
                <apex:inputField value="{!theCase.description}"
style="width:500px;height:92px;margin-top:4px;" />
                <div class="custompubblock" style="margin-top:5px;">
                    <div>Status: </div><apex:inputField value="{!theCase.status}"
/>   
                   <div>Priority: </div><apex:inputField value="{!theCase.priority}"
 />   
                  <div>Case Origin: </div><apex:inputField value="{!theCase.origin}"
 />
               </div>
                <div style="color:red;">{!lastError}</div>
            </apex:outputPanel>
        </apex:form><br/>
        <button type="button" onclick="createCase();"
style="position:fixed;bottom:0px;right:0px;padding:5px 10px;
        font-size:13px; font-weight:bold; line-height:
        18px;background-color:#0271BF;background-image:-moz-linear-gradient(#2DADDC,
#0271BF);background-repeat:repeat-x;
        border-color:#096EB3;" id="addcasebutton">Create Case</button>
    </div>
</apex:page>
```

Requirements for Refreshing Host Pages

To have an object-specific or global custom action refresh the feed on the page that hosts it, the Visualforce page you create to use as that action must:

- Reference the publisher JavaScript file: `<script type='text/javascript' src='/canvas/sdk/js/publisher.js'/>`. (Creating custom Visualforce actions doesn't require the Force.com Canvas SDK.)
- Include this JavaScript call: `Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed:true}});`.

SEE ALSO:

Visualforce Pages as Object-Specific Custom Actions

Create Global Quick Actions

Actions

Hide the Action Header for Visualforce Custom Actions

When creating a Visualforce page to use as a custom action, you can choose to hide the action's header. Hiding the action header helps prevent user confusion, especially if you have your own buttons specified in the Visualforce page.

To hide the header, add `showQuickActionVfHeader="false"` to the `<apex:page>` tag of the custom action's Visualforce page. When the Visualforce custom action renders in Salesforce1, the header and the Cancel and Save buttons are hidden. Using this attribute doesn't affect how the action displays in the full Salesforce site.

If you don't specify `showQuickActionVfHeader`, its value defaults to `true`.

The `showQuickActionVfHeader` attribute isn't supported in communities.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

SEE ALSO:

Visualforce Pages as Object-Specific Custom Actions

Visualforce Pages as Global Custom Actions

Prerequisites for Using Canvas Apps as Custom Actions

Using canvas apps as custom actions makes it easy to give users access to the functionality of your apps in Chatter and elsewhere in Salesforce.

You can use as a custom action any canvas app that uses Publisher as a location. For example, you might use an expense report app as a custom action to make it easy for salespeople to submit expense reports directly from feeds. A custom action that includes a video conferencing canvas app could help support agents communicate with customers visually for easier troubleshooting of technical issues.

Before creating a custom action with a canvas app, be sure the app uses Publisher as a location, and be sure to give the users you want to be able to use the action access to the app.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

- Custom Quick Actions
- Visualforce Pages as Object-Specific Custom Actions
- Actions

Enable Actions in the Chatter Publisher

Enabling actions in the publisher lets you add actions you've created to Chatter publishers on the Home page, on the Chatter tab, in Chatter groups, and on record detail pages.

By default, the Chatter publisher includes the standard actions Post, File, Link, Poll, Question, and Thanks. With the actions in the publisher setting enabled, you can include nonstandard actions in the Chatter publisher too. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

In organizations created after the Winter '14 release, actions in the publisher is enabled automatically.

1. From Setup, enter *Chatter Settings* in the Quick Find box, then select **Chatter Settings**.
2. Click **Edit**.
3. Select Enable Actions in the Publisher.
4. Click **Save**.

Note: You don't need to enable actions in the publisher in order to use them in Salesforce1 or in third-party apps. See Actions with and without Chatter for more information.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To enable actions in the Chatter publisher:

- "Customize Application"

IN THIS SECTION:

Set Up Actions with Chatter Enabled

Global and object-specific actions enhance the Chatter experience for your users in Salesforce Classic and in Salesforce1.

Set Up Actions Without Chatter Enabled

You can set up object-specific and global actions for Salesforce1 or third-party apps, even if your organization doesn't have Chatter enabled.

Actions with and without Chatter

Use actions regardless of whether Chatter or actions in the publisher are enabled.

Set Up Actions with Chatter Enabled

Global and object-specific actions enhance the Chatter experience for your users in Salesforce Classic and in Salesforce1.

Chatter must be enabled for your organization.

By default, the Chatter publisher in Salesforce Classic includes the standard actions Post, File, Link, Poll, Question, and Thanks. To set up more actions in Chatter:

1. Enable feed tracking for the objects for which you want to make actions available.
2. Enable actions in the publisher if you want to see both standard actions and non-standard actions in the Chatter publisher.
3. Optionally, enable feed updates for related records to display feed items on a record detail page when related records are created.
4. Create object-specific actions or global actions.
5. Customize the action layout with the fields that you want users to see when they use the action.
6. Add the actions to page layouts or global publisher layouts.

   Salesforce automatically adds default actions to the page layouts for account, case, contact, lead, and opportunity, and to the global publisher layout in organizations that were created after Winter '14.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To set up actions:
- "Customize Application"

SEE ALSO:

Enable Actions in the Chatter Publisher

Create Object-Specific Quick Actions

Set Up Actions Without Chatter Enabled

You can set up object-specific and global actions for Salesforce1 or third-party apps, even if your organization doesn't have Chatter enabled.

When Chatter is disabled in an organization, only the nonstandard actions appear in the action bar in Salesforce1 or in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

Follow these steps to set up actions for use in Salesforce1 or third-party apps.

1. Create object-specific actions or global actions.
2. Customize the action layout with the fields that you want users to see when they use the action.
3. Add the actions to page layouts or global publisher layouts.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To set up actions:
- "Customize Application"

Salesforce automatically adds default actions to the page layouts for account, case, contact, lead, and opportunity, and to the global publisher layout in organizations that were created after Winter '14.

SEE ALSO:

Create Object-Specific Quick Actions

Create Global Quick Actions

Customize Actions with the Enhanced Page Layout Editor

Actions with and without Chatter

Use actions regardless of whether Chatter or actions in the publisher are enabled.

The actions that are available in the full Salesforce site or in Salesforce1 To enable or disable Chatter for your organization, from Setup, enter *Chatter Settings* in the Quick Find box, then select **Chatter Settings**. If Chatter is enabled, the Enable Actions in the Publisher option controls whether the actions that you create display in the Chatter publisher.

	Chatter Off, Actions Off	Chatter On, Actions Off	Chatter On, Actions On
You can create global actions and customize global action lists	Yes	Yes	Yes
You can create object-specific actions and customize object-specific action lists	Yes	Yes	Yes
Actions appear on the Home page and Chatter home page in the full Salesforce site	No	Yes[1]	Yes
Actions appear in object feeds in the full Salesforce site	No	Yes[1,2]	Yes[2]
The action bar is available in the Salesforce1 feed	No[3]	Yes[4]	Yes
The action bar is available on the record view in Salesforce1	Yes[5]	Yes[6]	Yes[6]
The action bar is available on Lightning Pages in Salesforce1	Yes[5]	Yes	Yes

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Footnotes:

1. If actions in the publisher aren't enabled, only standard Chatter actions (Post, File, Link, Poll, and Thanks) appear in the Chatter publisher in the full Salesforce site.
2. The Chatter feed appears on an object's detail page in the full Salesforce site only for objects that have feed tracking enabled.
3. When Chatter is disabled, the Feed item isn't available in Salesforce1.
4. When Chatter is enabled but actions in the publisher aren't, standard Chatter actions and nonstandard actions appear in the Salesforce1 action bar and in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.
5. When Chatter and actions in the publisher are disabled, only nonstandard actions appear in the action bar in Salesforce1 or in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.
6. If feed tracking isn't enabled on the object, only nonstandard actions appear in the Salesforce1 action bar and in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

SEE ALSO:

Set Up Actions with Chatter Enabled

Set Up Actions Without Chatter Enabled

Actions

Action Layout Editor

Just as object record pages have page layouts that can be customized, actions have action layouts that can be customized. When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they're taking the action.

For example, a Post action needs just one large text area for user input. By contrast, an object-related Create action must include multiple fields; otherwise, Salesforce can't create the record.

The first time you view the layout for an action you've created, certain fields are prepopulated: target object default fields, standard required fields, and any custom universally required fields. Default actions (available in organizations created after Winter '14) have predefined sets of fields.

The upper part of the editor contains a palette, and below the palette is the action layout. The palette contains fields from the action's target object that you can add to the action layout, except for the following unsupported field types:

- Record type fields
- Read-only field types such as roll-up summary, formula, and auto-number fields
- Read-only system fields such as `Created By` or `Last Modified By`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Inactive Fields

Fields that are already on the action layout still appear on the palette but are inactive. When you select an inactive field on the palette, Salesforce highlights the field on the action layout.

Field Type Conversion

If you convert a field's type from one that is supported for actions to a type that isn't supported, Salesforce removes the field from the action layout. If you convert the field back to a supported type without changing the action layout, Salesforce automatically adds the field back to the layout. If you edit the layout and then convert the field back to a supported type, add the field back to the layout manually.

Layouts Used for Log a Call Actions

A Log a Call action takes the active task page layout except under the following conditions:

- Suppose that your organization has a custom Log a Call action for an object. The custom action takes the custom action layout defined for it.
- Now suppose that your organization has a custom Log a Call global action. That action takes the custom layout defined for it, unless you also have a custom Log a Call action for an object. (A custom action on an object overrides a custom global action.)

To display the simpler New Task form to Salesforce1 users, enable the form in Activity Settings and ensure that the layout used includes a subject field.

Layout Auditing

Salesforce tracks action layout customization in the setup audit trail history.

To view and edit the layouts for global actions in Setup, enter *Actions* in the Quick Find box, then select **Global Actions** and then click **Layout** next to the action's name. To view and edit the layouts for object-specific actions, find the object in Setup, then go to Buttons, Links, and Actions.

IN THIS SECTION:

Action Layout Editor Considerations

When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they're taking the action.

SEE ALSO:

Action Layout Editor Considerations

Actions

Action Layout Editor Considerations

When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they're taking the action.

- There is no hard limit to the number of fields you can add to an action layout. However, for optimum usability, we recommend a maximum of 8 fields. Adding more than 20 fields can severely impact user efficiency. To reduce the number of fields in your layout, you can create predefined values for the required fields, and then remove those fields from your layout. You can set predefined field values from the action detail page.
- Mobile smart actions are populated with all your organization's required fields on the relevant object, regardless of how many fields there are. For example, the New Case action in the mobile smart action bundle includes all required case fields. You can't edit the fields on mobile smart actions. The fields that appear change only if you change which fields on an object are required.
- You can remove a required field from the action layout, but make sure that the field has a predefined value. Otherwise, users can't create records.
- Rich text area fields are supported only when you add them to one-column layouts, or as fields that span both columns in two-column layouts. If you add a rich text area field to only one column in a two-column layout, it appears as a plain text area, because there's not enough space to display the full rich text editor.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

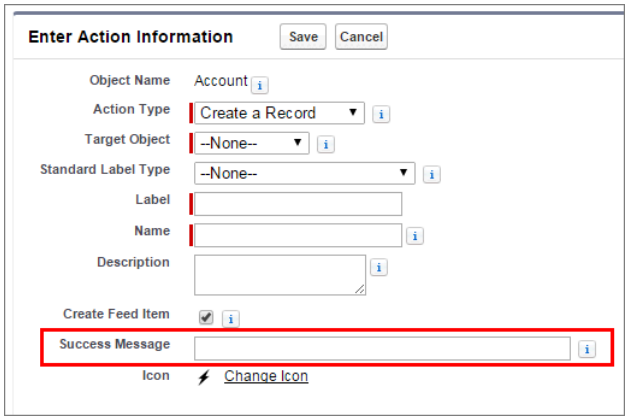
Custom Success Messages for Quick Actions

For Create a Record, Update a Record, and Log a Call action types, you can create a custom message that displays when the action executes successfully.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions



In Salesforce1 and Lightning Experience, when a create, update, or Log a Call action is completed, a default success message displays, regardless of whether the action created a feed item. If you add a custom success message to one of these actions, your custom success message displays instead of the default message.

In Salesforce Classic, custom success messages have slightly different behavior. If you select `Create Feed Item` for a Create a Record or Log a Call action, no success message displays in Salesforce Classic. The feed item that is created is the confirmation that the action executed successfully.

Note: If you have `All Related Objects` selected under feed tracking for a particular object, when a quick action creates related objects, a feed item is always created, regardless of the status of `Create Feed Item`.

SEE ALSO:

- Create Object-Specific Quick Actions
- Create Global Quick Actions

Customize Actions with the Enhanced Page Layout Editor

Use the page layout editor to customize which actions show up in Salesforce and in Salesforce1.

Tip: To manage the actions for global pages, such as Home, Chatter Home, and Chatter groups, see Global Publisher Layouts.

From the management settings for the object whose actions you want to manage, go to Page Layouts.

You can add actions to two sections on a page layout:

Quick Actions in the Salesforce Classic Publisher
This section can contain actions only from the Quick Actions category in the palette. Actions in this section appear in the Chatter publisher in Salesforce Classic.

Salesforce1 and Lightning Experience Actions
This section can contain actions only from the Salesforce1 & Lightning Actions category in the palette. On object page layouts, the Salesforce1 & Lightning Actions category contains all available types of actions for the object, including quick actions, productivity actions, Lightning component actions, and standard and custom buttons. Actions in this section appear in the action bar and action menu in Salesforce1 and in various areas of Lightning Experience.

Tip: Hover over an action in the palette to see its label, API name, and action type.

- To override the action defaults for an action section that you haven't customized, either click the override text or hover over the section and click .

  If you haven't customized the Quick Actions in the Salesforce Classic Publisher section of a page layout, the actions that appear in the publisher for that object default to the actions that are assigned to the global publisher layout. Upon overriding, the actions default to the standard actions—Post, File, Link, Poll, Question, and Thanks—regardless of what actions were assigned to the global publisher layout.

  If you haven't customized the Salesforce1 and Lightning Experience Actions section of a page layout, the actions for that object default to a set of predefined actions. If you have customized actions in the Quick Actions in the Salesforce Classic Publisher section, and have saved the layout, the Salesforce1 and Lightning Experience Actions section inherits the actions from the Quick Actions in the Salesforce Classic Publisher section, plus any standard or custom buttons present on the layout, when you click to override.

- To revert the actions in either section to the defaults for that section, hover over the section and click .

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To create actions:
- "Customize Application"

To customize action layouts and page layouts:
- "Customize Application"

To view page layouts:
- "View Setup"

SEE ALSO:

Set Up Actions with Chatter Enabled
Actions in Lightning Experience
Mobile Smart Actions
Customize Page Layouts with the Enhanced Page Layout Editor
Find Object Management Settings

Set Predefined Field Values for Quick Action Fields

When you create actions, use predefined field values to set a value for a field. Predefined values can help ensure consistency and make it faster and easier for users to create records.

When you configure action layouts, it's better to use fewer fields. Most users, especially mobile users, don't like to fill in a lot of required fields. They want to get things done and move on to their next task. A good way to use fewer fields in action layouts is to set predefined values for as many fields as possible. The more fields you can set predefined values for, the more you can remove from the layout and make the action easier and quicker to use. Balance ease of use with the need for required information. However, don't remove required fields from an action layout without setting a predefined value for those fields, or when a user applies that action, the record won't save properly.

If you set predefined values for fields on object records created through an action, you don't need to add those fields to the action layout. For example, when you configure an action that lets users create opportunities, set Prospecting as the predefined value for the `Stage` field. All new opportunities users create through that action are automatically assigned to the prospecting stage. You can remove the `Stage` field from the action's layout, because the field is going to be assigned a value automatically.

Tip: Predefined values for fields on actions are different from default values that you can set for fields on records. If a field is included in an action, it can have both a predefined value set for the action *and* a default value set.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To set predefined field values:

- "Customize Applications"

To set predefined field values:

1. Click the name of an action in the Buttons, Links, and Actions list or the Global Actions list.
2. On the action detail page, click **New** in the Predefined Field Values list.
3. Select the field you want to predefine a value for.
4. Specify the value for the field.
5. Click **Save**.

Tip: On object-specific actions, the predefined value can include references to the source object and its related objects.

IN THIS SECTION:

Notes on Predefined Field Values for Quick Actions

SEE ALSO:

Notes on Predefined Field Values for Quick Actions

Notes on Predefined Field Values for Quick Actions

Setting predefined field values for quick actions is especially important if you remove required or `Always on Layout` fields from the action layout.

- You can set predefined values for any field available in the action layout editor, with these exceptions.
  - Rich text area fields
  - Multi-select picklists
  - Read-only field types like auto-number, formula, and roll-up summary fields
- You can't use a dependent picklist to set a predefined value.
- If a field on an action has both a predefined value and a default value set, the action uses the predefined value, not the default value.
- If you set a predefined value for a field and leave it on the action layout, the predefined value displays as a default value for the field.
- If you have a required field with a predefined value assigned, and have removed the field from the action layout, if you later delete the field's predefined value you will need to re-add the required field to the action layout. Otherwise, users won't be able to save the record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

SEE ALSO:

Set Predefined Field Values for Quick Action Fields

Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu with Log A Call actions and Create a Record quick actions that point to the Events, Notes, and Tasks objects. And in Salesforce1, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

In Salesforce for Outlook, global publisher layouts drive the actions that Group, Contact Manager, and Professional Edition users see when they click the Salesforce Side Panel Publisher. Salesforce for Outlook users working in all other editions can set up their side panel publishers using Outlook Side Panel Publisher Layouts.

Note: Chatter groups without customers display the global publisher layout by default, unless you override it with a customized group publisher layout. In Chatter groups that allow customers, the publisher displays standard actions only, such as Post, File, Link, and Poll.

You can assign different global publisher layouts to different user profiles to customize which actions users see by default on global pages.

These are the steps involved in working with global publisher layouts.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To create actions:
- "Customize Application"

To customize action layouts and page layouts:
- "Customize Application"

To view page layouts:
- "View Setup"

IN THIS SECTION:

1. Create Global Publisher Layouts
   Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu with Log A Call actions and Create a Record quick actions that point to the Events, Notes, and Tasks objects. And in Salesforce1, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.
2. Add Actions to Global Publisher Layouts
   Actions you add to the global publisher layouts appear on pages such as the Home and Chatter pages, and the action bar and action menu on the Feed and People pages in Salesforce1.
3. Assign Global Publisher Layouts to User Profiles
   Once you finish creating a global publisher layout, you can assign it to different user profiles. For example, a Marketing User and a Standard User might need different actions in the Chatter publisher or in Salesforce1. Create multiple global publisher layouts and assign them to different user profiles to customize the actions for each profile.

SEE ALSO:

Create Global Quick Actions

Create Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu with Log A Call actions and Create a Record quick actions that point to the Events, Notes, and Tasks objects. And in Salesforce1, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

1. From Setup, enter *Publisher Layouts* in the Quick Find box, then select **Publisher Layouts**.
2. To create a new global publisher layout, click **New**.
3. To clone a publisher layout, select one from the Existing Global Publisher Layout drop-down list.
4. Enter a name for the new global publisher layout.
5. Click **Save**.

SEE ALSO:

Assign Global Publisher Layouts to User Profiles

Global Publisher Layouts

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To create actions:
- "Customize Application"

To customize action layouts and page layouts:
- "Customize Application"

To view page layouts:
- "View Setup"

Add Actions to Global Publisher Layouts

Actions you add to the global publisher layouts appear on pages such as the Home and Chatter pages, and the action bar and action menu on the Feed and People pages in Salesforce1.

Arrange the actions so that the frequently used actions appear first in each list.

Walk Through It: Customizing Global Publisher Layouts

You can add actions to two sections on a page layout:

Quick Actions in the Salesforce Classic Publisher
This section can contain actions only from the Quick Actions category in the palette. Actions in this section appear in the Chatter publisher in Salesforce Classic.

Salesforce1 and Lightning Experience Actions
This section can contain actions only from the Salesforce1 & Lightning Actions category in the palette. On object page layouts, the Salesforce1 & Lightning Actions category contains all available types of actions for the object, including quick actions, productivity actions, Lightning component actions, and standard and custom buttons. Actions in this section appear in the action bar and action menu in Salesforce1 and in various areas of Lightning Experience.

1. From Setup, enter *Publisher Layouts* in the Quick Find box, then select **Publisher Layouts**.
2. To add or remove actions, drag them to and from the palette. To reorder actions, select an action and drag it to a new position.
3. Click **Save** when you're done, or click **Quick Save** to save your changes and continue working on the layout.

   If you navigate away without saving, you lose your changes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

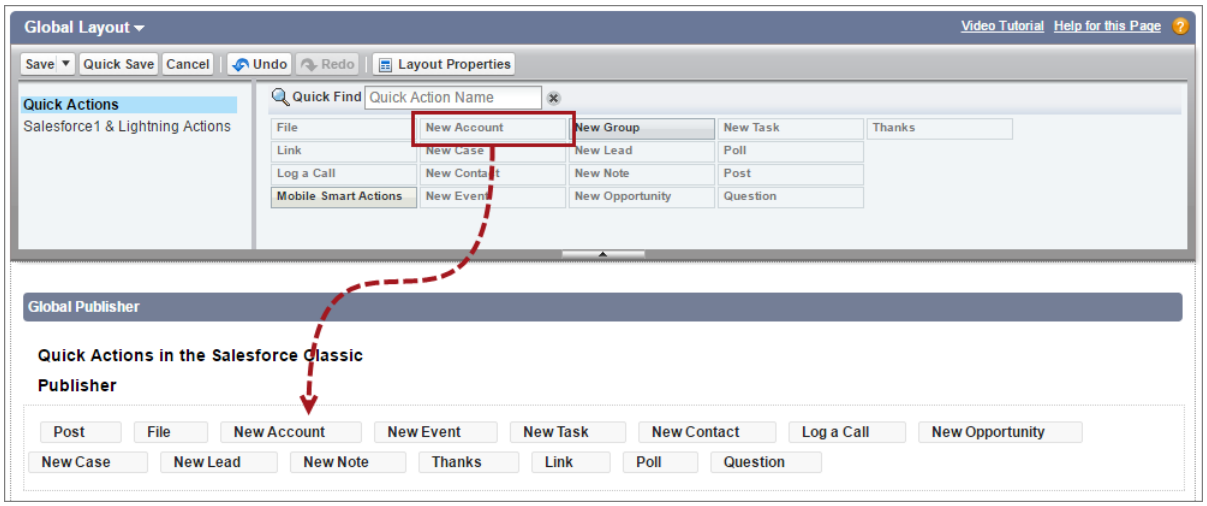
USER PERMISSIONS

To create actions:
- "Customize Application"

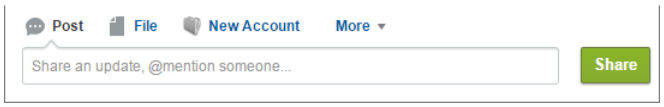
To customize action layouts and page layouts:
- "Customize Application"

To view page layouts:
- "View Setup"

Example: Let's add the New Account action to the publisher on the Home and Chatter pages in Salesforce Classic, so users can create an account directly from the publisher. Drag the **New Account** action to the Quick Actions in the Salesforce Classic Publisher section and save your changes.



Go to the Chatter tab in Salesforce Classic. Now the New Account action shows up in the publisher.



Note: The Chatter page in Lightning Experience supports only the standard Chatter actions Post, Poll, and Question, and if you have Groups, the Announcement action.

SEE ALSO:

Actions in Lightning Experience

Assign Global Publisher Layouts to User Profiles

Assign Global Publisher Layouts to User Profiles

Once you finish creating a global publisher layout, you can assign it to different user profiles. For example, a Marketing User and a Standard User might need different actions in the Chatter publisher or in Salesforce1. Create multiple global publisher layouts and assign them to different user profiles to customize the actions for each profile.

1. From Setup, enter *Publisher Layouts* in the Quick Find box, then select **Publisher Layouts**.
2. Click **Publisher Layout Assignment**.
3. Click **Edit Assignment**.
4. Select a user profile by clicking anywhere on its row in the table.
5. From the Publisher Layout to Use drop-down, select the global publisher layout that you want to assign to the highlighted profile.
6. Save the layout.

SEE ALSO:

Add Actions to Global Publisher Layouts

Global Publisher Layouts

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

USER PERMISSIONS

To create actions:
- "Customize Application"

To customize action layouts and page layouts:
- "Customize Application"

To view page layouts:
- "View Setup"

Actions in Lightning Experience

In Lightning Experience, actions display in the Global Actions menu in the header, on list view items, and in several places on a record page. Where they display on a record page depends on the action's type.

EDITIONS

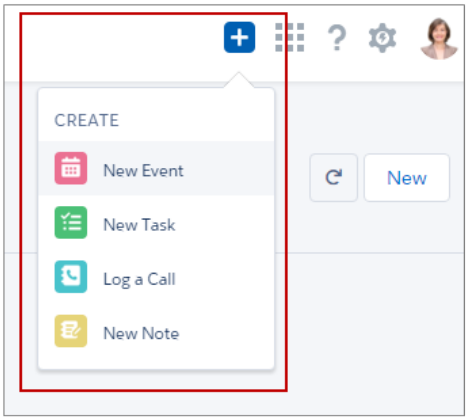
Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Actions in the Global Actions Menu

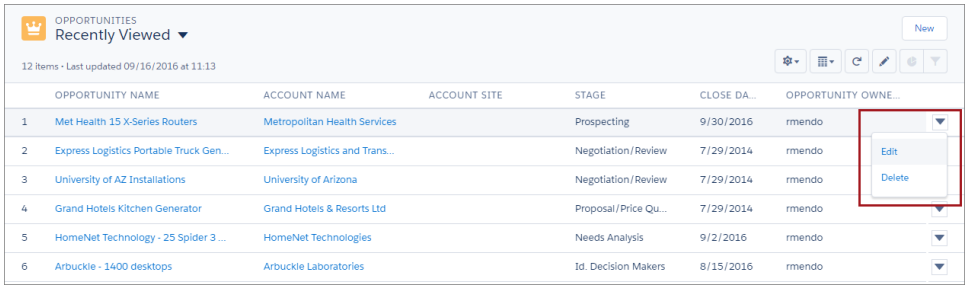
The Global Actions menu displays a subset of global actions from the Salesforce1 and Lightning Experience Actions section of the global publisher layout. Specifically, these types of global actions are supported in the menu: Log a Call and Create a Record quick actions that point to the Events, Notes, and Tasks objects.



The items in the menu display in the order that they're listed in the Salesforce1 and Lightning Experience Actions section of the page layout.

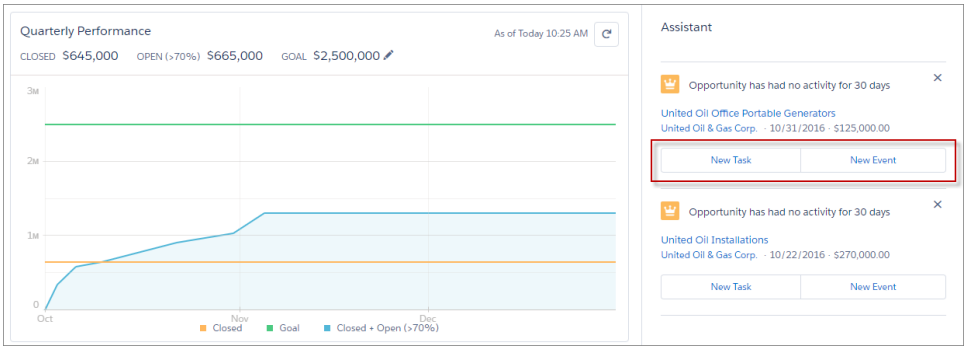
Actions on List View Items

Except for the Tasks object, only standard button actions are supported on list view items. Items in Tasks list views contain the full list of actions available for tasks.



Actions on the Home Page

On the Home page, you can find actions on recommendations in the Assistant. For example, if a sales rep receives an update that an opportunity doesn't have any open activity, the rep can create a task or event directly from the recommendation.



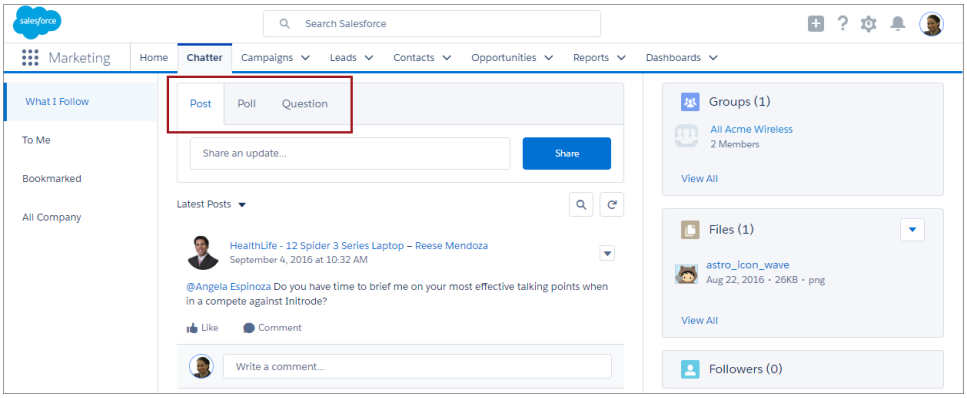
The actions that appear depend on the type of recommendation. The available actions include:

- New Task
- New Event
- Edit
- Email

After you complete an action, the related recommendation disappears from the Assistant.

Actions on the Chatter Page

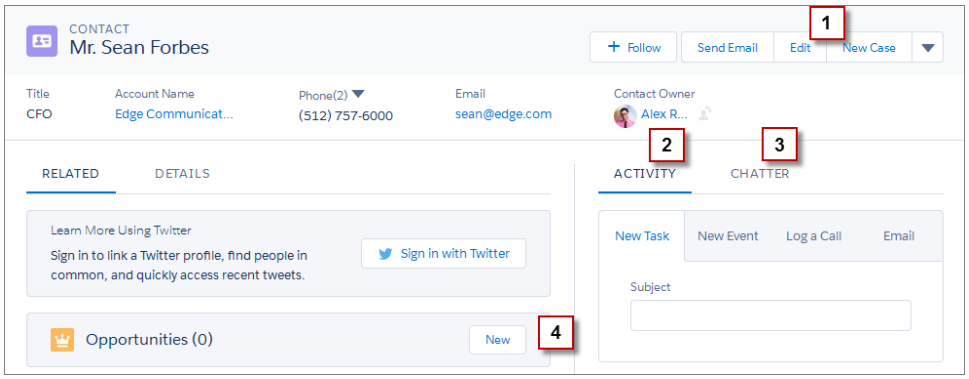
The Chatter page, like the Chatter tab on record pages, contains only standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. You can add, remove, or reorder the actions on the Chatter page from the Salesforce1 and Lightning Experience Actions section of the global publisher layout.



Actions on Record Pages

Here's a sample contact page in Lightning Experience.

Note: The opportunity and leads workspaces have different structures, but actions appear in the same way on those pages.



The page-level action menu in the record's highlights panel (**1**) contains:

- Productivity actions
- Global and object-specific quick actions, except for those related to creating tasks, creating events, and logging calls
- Standard buttons
- Custom Lightning component quick actions
- Custom Visualforce quick actions
- Custom Visualforce buttons
- Canvas actions

The actions that appear in the page-level action menu display in the order that they are listed in the Salesforce1 and Lightning Experience Actions section of the page layout.

Note: Custom buttons that call JavaScript aren't supported in Lightning Experience.

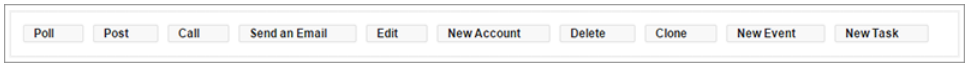
The Activity tab (**2**) contains Log A Call actions and Create a Record quick actions that point to the Event and Task objects. It also contains the Email standard button.

The Chatter tab (**3**) contains standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. Some objects support other standard Chatter actions predefined by Salesforce.

Actions on Related Lists

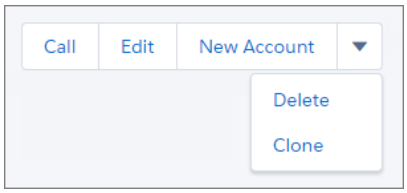
Related lists (**4**) contain the standard buttons normally found on related lists. Usually, it's simply the New button.

Example: Let's say you have these actions on your Contact page layout in the Salesforce1 and Lightning Experience Actions section.

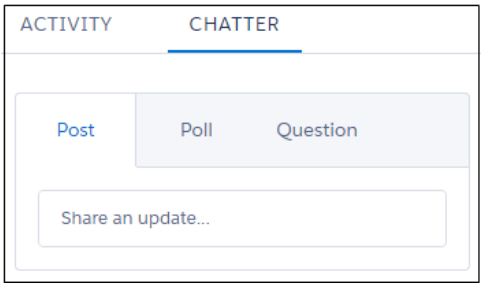


You have quick actions (New Account, New Event, New Task), a productivity action (Call), standard buttons (Edit, Delete, Clone, Send an Email), and Chatter actions (Poll, Post). Here's how those actions display on a contact record page in Lightning Experience.

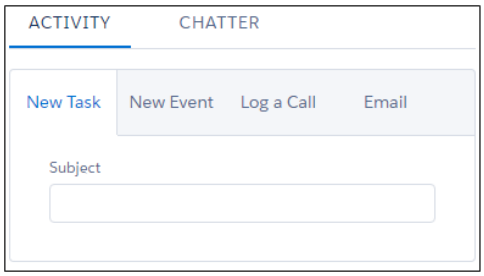
- The actions in the page-level action menu are a combination of the quick actions, productivity actions, and standard buttons in the order that they're listed on the page layout. Although they're quick actions, New Event and New Task don't show up here.



- The Chatter actions from the front of the action list are on the Chatter tab.



- The Activities-related actions—Email, New Event, New Task—display on the Activity tab.



IN THIS SECTION:

How Actions Are Ordered in Lightning Experience

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce1 and Lightning Experience Actions section of the page layout for that object. The same section on global publisher layouts determines the global actions that appear in the Global Actions menu.

SEE ALSO:

How Actions Are Ordered in Lightning Experience

Actions

How Actions Are Ordered in Lightning Experience

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce1 and Lightning Experience Actions section of the page layout for that object. The same section on global publisher layouts determines the global actions that appear in the Global Actions menu.

If you haven't customized the Salesforce1 and Lightning Experience Actions section of an object's page layout, the quick actions that appear on the object's record pages are derived from:

- The actions on the global publisher layout
- Standard and custom buttons in the buttons section of the object page layout

If you customize the Salesforce1 and Lightning Experience Actions section, the standard and custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the buttons as actions from the Salesforce1 & Lightning Actions category in the palette.

The actions in each section of the record page respect the ordering of its types of actions on the page layout.

The Global Actions menu ( ) in the Lightning Experience header displays these global actions from the Salesforce1 and Lightning Experience Actions section on the global publisher layout: Log A Call actions and Create a Record quick actions that point to the Events, Notes, and Tasks objects.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Actions in Lightning Experience

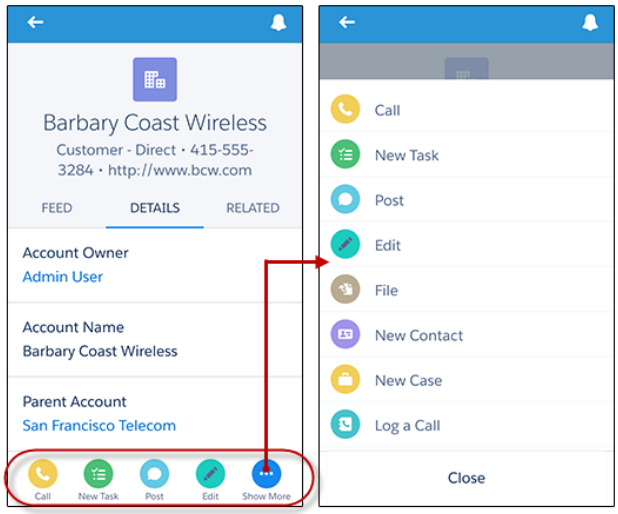
Actions

Salesforce1 Action Bar

Salesforce1 users have a one-stop place to find actions, so there's no confusion about where to go to do something. The action bar and its associated action menu collect actions from different places in Salesforce1 into a single, unified home.

In Salesforce1, productivity actions, standard and custom record buttons, and quick actions appear in the action bar and the action menu ( ). The action bar and action menu show all the available actions for a given page.

Salesforce1 Actions in the Action Bar and Action Menu



The action bar appears in most places in the app, including the feed, groups, user profiles, dashboards and reports, standard and custom object record views, related lists, and search results. The actions that are available depend on where a user is in Salesforce1 and on how you've configured page layouts and publisher layouts for your organization.

Users may see some or all of these kinds of actions in the action bar (including the action menu).

- Productivity actions—The Send Email ( ), Call ( ), Map ( ), View Website ( ), and Read News ( ) actions are available on accounts, contacts, leads, and person accounts. The Quick Message ( ), Join Conference Call ( ), and Map ( ) actions are available on mobile calendar events in Salesforce Today.

  Tip: A productivity action displays only if a record includes the information that the action is keyed to. For example, the Send Email action depends on the record including an email address. The View Website action requires the record to include a website URL.

- Custom and standard buttons—Buttons (such as Edit, Delete, or Clone) that are included in the Buttons section on an object's page layout are available in Salesforce1 as actions in the action bar on record pages. If you haven't customized the action order, the button order in the button section of the page layout is used. However, the Edit button is in a fixed position.

  Note: Custom links, custom buttons that are added to list views, and custom buttons that define the content source as `OnClick JavaScript` aren't supported and don't appear in Salesforce1.

- Quick actions—If you add, remove, or reorder actions in the action bar in the global publisher layout or an object's page layout, the changes are reflected in Salesforce1.
- Standard Chatter actions—Actions unique to Chatter, such as Post, File, Link, or Poll.

What are the exceptions?

- On record feeds, the **Follow**/**Unfollow** button remain in the highlights area on the page.
- On public and private group feeds, the **Join Group** and **Ask to Join** buttons remain in the highlights area, but the **Leave Group** button is in the action bar.

What does this mean for custom branding?

There can be only one version of a custom icon. Custom action icons that you created before Winter '15 are still supported, but they are truncated in the action bar. To optimize your custom action icons for display in the action bar, use these guidelines.

- The icon should be 72 x 72 pixels. Use the full pixel area for the image—don't leave spacing around the image like before.
- Make the image a PNG with a transparent background, with a file size that is less than 10k.
- Have a resolution of 72 dpi.
- Make the icon graphic white or lighter than the background color.
- Avoid heavy inner or outer shadows.
- Use simple and flat styling resembling the Salesforce1 icon family.

IN THIS SECTION:

List Item Actions in Salesforce1

List item actions is a version of the action bar that's available for items in list views, task lists, and related record lists. It enables Salesforce1 users to update records directly from lists without opening them.

How Actions Are Ordered in the Salesforce1 Action Bar

The Salesforce1 and Lightning Experience Actions section of a page layout and global publisher layout drives which actions appear in the Salesforce1 action bar. It also enables you to customize the order of quick actions, productivity actions, and standard and custom buttons that are available as actions.

How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions

Your org's page layouts and publisher layouts control the order in which actions appear in the Salesforce1 action bar and list item actions. If you don't customize the actions in the action bar on a page layout or global publisher layout, the location of key actions is predefined by Salesforce.

Considerations for Actions in Salesforce1

SEE ALSO:

How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions

How Actions Are Ordered in the Salesforce1 Action Bar

List Item Actions in Salesforce1

Considerations for Actions in Salesforce1

List Item Actions in Salesforce1

List item actions is a version of the action bar that's available for items in list views, task lists, and related record lists. It enables Salesforce1 users to update records directly from lists without opening them.

To access list item actions, navigate to a list view or task list or open a related list from an object's related information page. Then swipe left on the desired record.

Hide list item actions by swiping the list item back to the right or by tapping another item in the list.

List Views

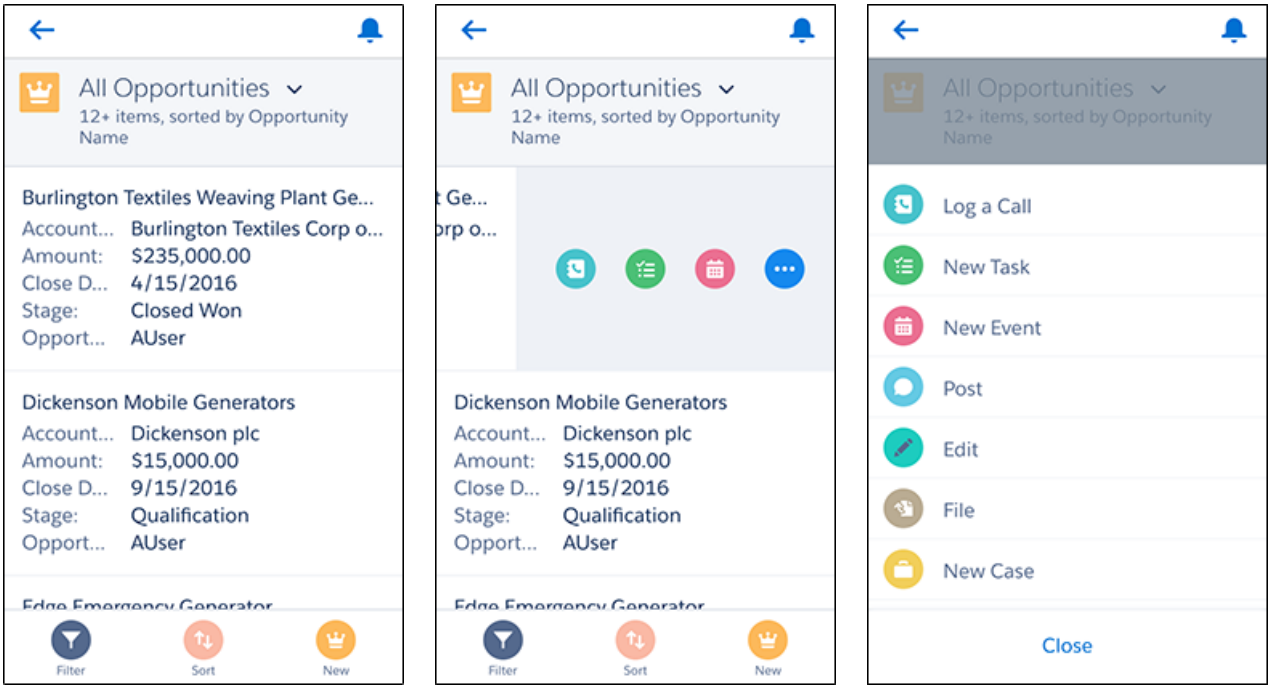
List item actions in list views don't have the same actions that are available in the action bar when viewing an object's record. For example, when a user visits an opportunity record in Salesforce1, the actions in its action bar reflect the actions in the Salesforce1 and Lightning Experience Actions section of the Opportunity page layout.

However, when the user swipes left on an opportunity from a list view, as you can see below, the actions that display come from the list of predefined actions for opportunities. See How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions for the breakdown of predefined actions for each object.

Here's the All Opportunities list view.

Swipe a list item to the left to reveal list item actions.

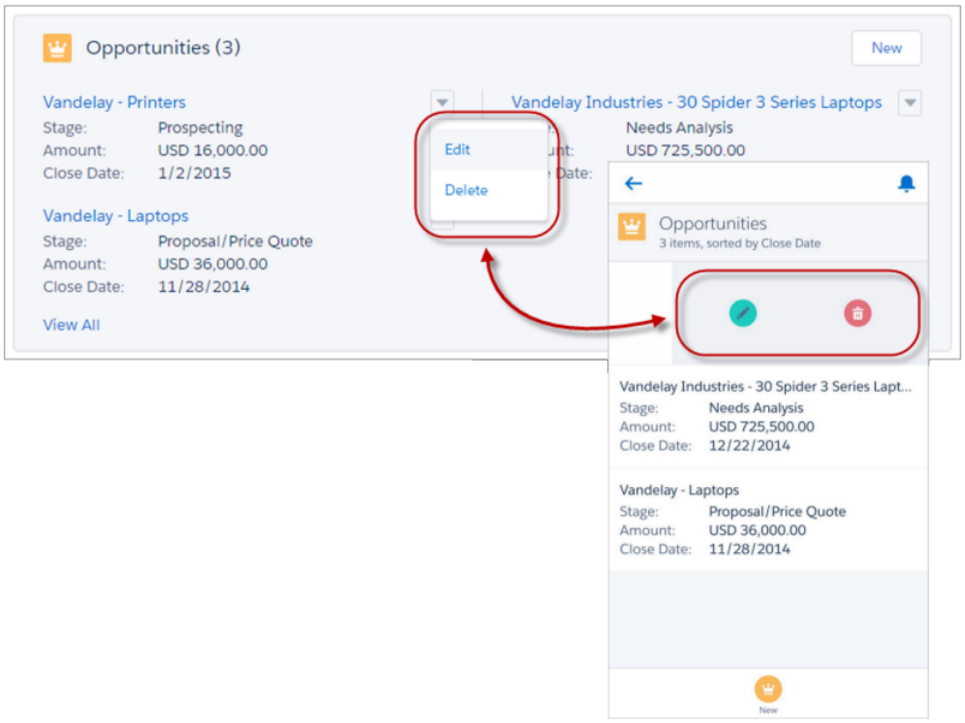
Tap to show the action menu, with the full list of actions available for the list item.



Related Lists

List item actions on related lists in Salesforce1 reflect the same actions as the related list in Lightning Experience. Usually, these are the standard actions Edit and Delete, but can vary by object. For example, in Lightning Experience, the actions available in the drop-down menu on an Opportunity related list item are the same set of actions that Salesforce1 users see when swiping left on the same record in the related list.

Note: Even if your org doesn't have Lightning Experience enabled, the actions on related lists that you see in Salesforce1 still match the actions that would appear on the related list items in Lightning Experience if it was enabled.



Note: Task and event list items are different than other objects when they display in related lists. In Salesforce1, when you tap the Tasks item from the menu, you can swipe left on an item in the My Tasks list view and see the predefined actions for tasks. However, tasks and events in the Open Activities or Activity History related lists in Salesforce1 have no actions and aren't swipe-able.

SEE ALSO:

- Salesforce1 Action Bar
- How Actions Are Ordered in the Salesforce1 Action Bar
- Considerations for Actions in Salesforce1

How Actions Are Ordered in the Salesforce1 Action Bar

The Salesforce1 and Lightning Experience Actions section of a page layout and global publisher layout drives which actions appear in the Salesforce1 action bar. It also enables you to customize the order of quick actions, productivity actions, and standard and custom buttons that are available as actions.

If you customize the Salesforce1 and Lightning Experience Actions section of a layout, Salesforce1 reflects your customizations.

If you customize the Quick Actions in the Salesforce Classic Publisher section, but not the Salesforce1 section, the actions in the Salesforce1 action bar are a combination of those in the Quick Actions in the Salesforce Classic Publisher section plus any standard or custom buttons present on the page layout.

If you customize the Salesforce1 and Lightning Experience Actions section, the standard and custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the buttons as actions from the Salesforce1 & Lightning Actions category in the palette.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

If neither section is customized, the action bar inherits a default set of actions predefined by Salesforce. The sets of actions differ between objects, based on the most common or typical activities required for each object.

SEE ALSO:

- How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions
- Salesforce1 Action Bar
- Considerations for Actions in Salesforce1

How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions

Your org's page layouts and publisher layouts control the order in which actions appear in the Salesforce1 action bar and list item actions. If you don't customize the actions in the action bar on a page layout or global publisher layout, the location of key actions is predefined by Salesforce.

Important: This predefined ordering chart applies to the Salesforce1 action bar only if you *haven't* customized the Salesforce1 and Lightning Experience Actions section of an object's page layout or on a global publisher layout.

The quick actions in the predefined set are derived from the actions in the "Quick Actions in the Salesforce Classic Publisher" section of the object page layout or global publisher layout.

On object page layouts, when the Salesforce1 section isn't customized:

If you've customized the actions in the Quick Actions in the Salesforce Classic Publisher section, the quick actions in the action bar reflect those customizations.

If neither section is customized, the quick actions you see in the action bar come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout.

On global publisher layouts, when the Salesforce1 section isn't customized:

If the Quick Actions in the Salesforce Classic Publisher section is customized, the quick actions in the action bar inherit those customizations.

If neither section is customized, the quick actions in the action bar for global pages default to a Salesforce predefined set.

The predefined actions in the action bar, list item actions, and associated action menus are divided into groups. The arrangement of these groups is fixed. The order of actions within the groups can vary based on the object and the actions that are present on the global publisher layout or on an object's page layout. Not every object or page displays every group.

Note: Actions on list view items reflect only the predefined set of actions for that object. For example, let's say you're viewing the All Accounts list in Salesforce1. If you swipe left on an account item in the list, you see a set of actions. Those actions come from the predefined list of actions for accounts in this chart. You always see Call, Edit, and Delete. The other actions on the list view item follow the order and rules defined for the action groups in the chart.

Here's the breakdown of which actions are contained in each group for each object or page.

Object or Page	Action Group 1	Action Group 2	Action Group 3	Action Group 4	Action Group 5	Action Group 6
Account	1. Call, 2. New Task, 3. New Event, 4. Post	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	Send Text (if the `Phone` field is populated) View Website (if the `Website` field is populated)
Case	Actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Edit		Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	
Contact	1. Call, 2. Send Email, 3. New Task, 4. New Event	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	Send Text

Object or Page	Action Group 1	Action Group 2	Action Group 3	Action Group 4	Action Group 5	Action Group 6
			Classic Publisher section of the *global publisher layout*.			
Custom Object	First four actions in the order defined on the page layout. If the Quick Actions in the Salesforce Classic Publisher section isn't customized, then the first four actions in the order defined on the *global publisher layout*.	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	
Event	Quick actions in the order defined on the layout. Standard Chatter actions aren't supported.	Edit, Delete				
Feed	Quick actions in the order defined on the global publisher layout					
Group	Actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the				Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	

Object or Page	Action Group 1	Action Group 2	Action Group 3	Action Group 4	Action Group 5	Action Group 6
	global publisher layout.					
Lead	1. Log a Call, 2. New Task, 3. Convert (if enabled), 4. Post	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	Call, Send Text, Send Email
"App Page" Lightning Page	Global actions in the order defined in the Lightning Page					
List View	New					
Object Home Page (Tablet Only)	1. New, 2. Sort					
Opportunity	1. Log a Call, 2. New Task, 3. New Event, 4. Post	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	

Object or Page	Action Group 1	Action Group 2	Action Group 3	Action Group 4	Action Group 5	Action Group 6
People	1. Call, 2. Send Email, 3. Post		Remaining actions in the order defined on the global publisher layout			
Person Account	1. Call, 2. Send Email, 3. New Task, 4. New Event	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	Map, Read News, Send Text, View Website
Related List (for standard objects)	1. New					
Salesforce Today—Main Page	Quick actions in the order defined on the global publisher layout					
Salesforce Today—Mobile Calendar Event	1. Quick Message, 2. Join Conference Call, 3. Map		Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the			

Object or Page	Action Group 1	Action Group 2	Action Group 3	Action Group 4	Action Group 5	Action Group 6
			global publisher layout.			
Task	1. Edit Comments, 2. Change Date, 3. Change Status, 4. Change Priority	5. Edit	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the *global publisher layout*. Standard Chatter actions aren't supported.	Custom buttons that are supported in Salesforce1, in the order defined on the page layout.*	Remaining standard buttons that are supported in Salesforce1, in the order defined on the page layout.	

As we mentioned, some actions are in fixed positions. In places where you see a numbered list in the table, this is the fixed order that those actions appear in on the action bar, list item actions, and in the respective action menus.

For example, for the Account object, the standard Chatter Post action is in the fourth position. This is fixed. Regardless of where you put the Post action in the account page layout, Post always displays in the fourth position.

However, deletion of actions is always respected. So in our example, if you delete the Post action from the account page layout, the remaining actions move up and you see Edit in the fourth position.

* Custom buttons that are added to the Button section of a page layout and that define the content source as `URL` or `Visualforce` are supported in Salesforce1. Remember that Visualforce pages must be enabled for use in Salesforce1. Custom links, custom buttons that are added to list views, and custom buttons that define the content source as `OnClick JavaScript` aren't available in Salesforce1.

SEE ALSO:

- How Actions Are Ordered in the Salesforce1 Action Bar
- Salesforce1 Action Bar
- Considerations for Actions in Salesforce1

Considerations for Actions in Salesforce1

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

- A productivity action displays only if a record includes the information that the action is keyed to. For example, the Send Email action depends on the record including an email address. The View Website action requires the record to include a website URL.
- If you're using record types in your organization, some quick actions might not be visible to your users. For more information, see Quick Actions and Record Types on page 902.
- The Mobile Smart Actions element appears as a single action element in the page layout editor, but it expands to several actions when it displays in Salesforce1. If the Mobile Smart Actions element is in the Quick Actions in the Salesforce Classic Publisher section, and you then customize the action bar section, the actions in Salesforce1 are derived from the action bar customizations you make. In this case, the Mobile Smart Actions element in the Quick Actions in the Salesforce Classic Publisher section becomes irrelevant.
- To customize the actions in the Salesforce1 action bar for standard and custom objects, first override the predefined actions. You can then add or remove actions from the Salesforce1 and Lightning Experience Actions section.

  For instance, to move the Join Group, Edit Group, or Leave Group actions on groups in Salesforce1, override the predefined actions in the Groups page layout.
- If you use URL custom buttons to pass parameters to standard pages in Salesforce Classic—such as pre-populating fields when creating a record—this behavior doesn't work in Salesforce1 or Lightning Experience.
- If you add a custom button to the Salesforce1 and Lightning Experience Actions section of a page layout, the icon for that custom button in the Salesforce1 action bar is a lightning bolt. If the Salesforce1 and Lightning Experience Actions section isn't customized, the icon of that custom button in the Salesforce1 action bar is a wrench.

SEE ALSO:

How Actions Are Ordered in the Salesforce1 Action Bar

How Predefined Actions Are Ordered in the Salesforce1 Action Bar and List Item Actions

Quick Actions and Record Types

Using record types in your organization can affect the availability of quick actions for your users.

If users don't have access to a particular record type, actions that are assigned to that record type aren't available to them. For example, let's say that you have a page layout that contains a mix of actions—some have no record type assigned and some are assigned to Record Type A. Users without access to Record Type A see only the nonassigned actions when they visit the page.

Important: Don't assign actions to the Master record type. The Master record type is a placeholder record type that's assigned when your organization is created.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Default Global Actions: A Special Case

If you have default global actions in your organization and you're using record types, your users might not be able to see all the default actions that are assigned to a page layout.

Default global actions are assigned to the Master record type, which isn't accessible to most profiles. As a result, default global actions with the Master record type that are associated with target objects that have record types configured aren't available for most users. To fix this issue, edit the default global actions associated with those objects and reassign them to a different record type.

For example, the New Contact default global action has Contact as its target object. Let's say you have record types set up for the Contact object, and you add the New Contact default global object to a page layout. Users who visit records based on that page layout don't see the New Contact action because the action is assigned to the Master record type by default. Editing the New Contact default global action and assigning it to a record type other than Master makes it available for all users who have access to its assigned record type.

Actions Best Practices

Use these tips as you set up actions to make the feature easy to use and easy to maintain.

EDITIONS

Actions available in: both Salesforce Classic and Lightning Experience

Publisher available in: Salesforce Classic

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Tips for Creating Actions

- Action labels longer than approximately 12–14 characters are shortened when they're displayed in the Chatter publisher. Keep names short and descriptive.
- Give your actions task-oriented names that tell your users what they do. Use terms such as New, Create, Share, Update, or Import.
- Use the `Description` field to create notes for yourself about each action. Notes are especially useful if you're creating several similar actions for different record types, for example. The description appears in the list of buttons, links, and actions for object-specific actions, or in the list of global actions, as well as on the detail page for the action. Your notes aren't visible to users.
- When creating custom actions that are going to be displayed in the Chatter publisher, limit their height to 400 pixels so that they are displayed correctly.

Tips for Laying Out Actions

- When customizing action layouts, consider what your users will do with them. Minimalism is key. Include only the fields that are necessary for them and for whomever handles the cases, calls, or records that result from those actions.
- To create a single-column layout, such as for display on mobile devices, add fields only in the left column.

- Use predefined field values to set standard values for common fields. For example, when you configure an action that lets users create opportunities, set Prospecting as the predefined value for the `Stage` field. All new opportunities users create through that action are automatically assigned to the prospecting stage. You can remove the `Stage` field from the action's layout, because the field is going to be assigned a value automatically. If you set predefined values for fields on object records created through an action, you don't need to add those fields to the action layout.
- Use the **Preview As...** button on the Action Layout Editor to see how an action layout appears to different user profiles.

Tips for Adding Actions to Publishers

- Because adding too many actions can cause the page to load slowly, we recommend including no more than nine actions total in each publisher, including any standard actions.
- In the full Salesforce site, if you include five or more actions in a publisher, three are shown and the rest are added to the publisher's **More** menu. If you include four or fewer actions, they're all shown.
- In Salesforce1, the first four actions show up in the action bar, and the full set of actions are accessed by tapping .

SEE ALSO:

Troubleshooting Actions

Set Up Actions with Chatter Enabled

Troubleshooting Actions

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

I don't see feeds on record detail pages for a certain object.

Feeds appear only for objects for which you've enabled feed tracking.

I see the feed on a record detail page, but I don't see a publisher.

If there are no actions in the Quick Actions in the Salesforce Classic Publisher section on a page layout, the publisher in Salesforce Classic doesn't appear. Add at least one action to the page layout for the publisher to appear.

I can create actions, but I can't add them to publishers.

Enable actions in the publisher to add nonstandard actions to publishers in Salesforce Classic.

I'm using Internet Explorer 10 and all the actions I've created appear in the publisher with the same icon, even though the actions are for different types of objects.

Internet Explorer version 10 doesn't support the techniques Salesforce uses to show icons that correspond to the type of object an action is associated with. Consider using Chrome, Firefox, Safari, or an earlier version of Internet Explorer.

I've added an action to a page layout, but a user assigned to the profile that uses that page layout can't see the action.

Be sure that the user has both Read and Edit permissions on the action's relationship field. The relationship field is the field that's automatically populated on the target object when a user creates a record using an action. For example, for an action on case that lets users create child cases, the default relationship field is `Parent Case`. To be sure users can see the Create Child Case action, check that they have both Read and Edit permissions on the `Parent Case` field.

I don't see a relationship field on my global create actions.

Relationship fields apply only to object-specific create actions, not to global actions.

I don't see some of the actions in my Chatter groups.

Which actions you see depends on your role in the group, the type of group, and how your administrator has configured the publisher layout for groups. Chatter groups without customers display the global publisher layout by default, unless you override it with a customized group publisher layout. In Chatter groups that allow customers, the publisher displays standard actions only, such as Post, File, Link, and Poll.

SEE ALSO:

- Actions Best Practices
- Set Up Actions with Chatter Enabled

Custom Buttons and Links

Custom buttons and links help you integrate Salesforce data with external URLs, applications, your company's intranet, or other back-end office systems.

Salesforce supports these custom links and custom buttons.

Type of Custom Link or Custom Button	Setup Page Where It's Configured
Bookmark-style links defined in the standard custom links home page component	Home Page Components
Full-featured custom links included in custom home page components	Custom Links
Full-featured custom links or custom buttons on objects	Buttons, Links, and Actions (in the object's management settings)

EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

Custom links can link to an external URL, such as `www.google.com`, a Visualforce page, or your company's intranet. Custom links can also link to a custom s-control in the custom s-control library, such as a Java applet or Active-X control.

Custom buttons can:

- Connect users to external applications, such as a web page that displays a map to a contact's address.
- Run an s-control from the s-control library, such as an s-control that escalates a case from the case detail page.
- Launch custom links.

You can choose the display window properties that determine how the target of a link or button is displayed to your users. Custom links and s-controls can include Salesforce fields as tokens within the URL or custom s-control. For example, you can include an account name in a URL that searches Yahoo: `http://search.yahoo.com/bin/search?p={!Account_Name}`.

You can override the default action of some standard buttons and customize the behavior of tab home pages to suit your org's needs.

A Web tab or custom link could display a blank page if the embedded site:

- Has been set to deny the loading of its content in a frame.
- Has been set to allow the loading of its content in a frame only if the same site is delivering the content.
- Contains a mix of secure and unsecure content, and the user's browser has been configured to block mixed active content.

To resolve this issue, try these workarounds.

- Set your custom link to either open in a new window or display in the existing window without the sidebar or header.
- Move the URL from a Web tab into a custom link instead, and set the URL to either open in a new window or display in the existing window without the sidebar or header.
- If the site you're embedding has an HTTP prefix and mixed active content, try changing the prefix to HTTPS. If the embedded site has a valid security certificate and it hasn't blocked itself from being displayed in frames, using HTTPS as the prefix allows the site to display.

IN THIS SECTION:

Define Custom Buttons and Links

Define the action that occurs when a user clicks a custom button or link. Custom buttons and links can streamline actions within Salesforce or integrate Salesforce data with external URLs, applications, or systems.

Override Standard Buttons and Tab Home Pages

You can override the behavior of standard buttons on record detail pages. You can also override the tab home page that displays when a user clicks a standard, custom, or external object tab.

Custom Button and Link Samples

Use samples of custom Salesforce buttons and links to determine whether they can work for you.

Custom Button and Link Considerations

Keep these considerations in mind when working with custom buttons and links.

Viewing References to Salesforce Components

SEE ALSO:

Define Custom Buttons and Links

Adding Default Custom Links

Customizing Home Tab Page Layouts

Define Custom Buttons and Links

Define the action that occurs when a user clicks a custom button or link. Custom buttons and links can streamline actions within Salesforce or integrate Salesforce data with external URLs, applications, or systems.

Watch a demo video of creating a custom link.

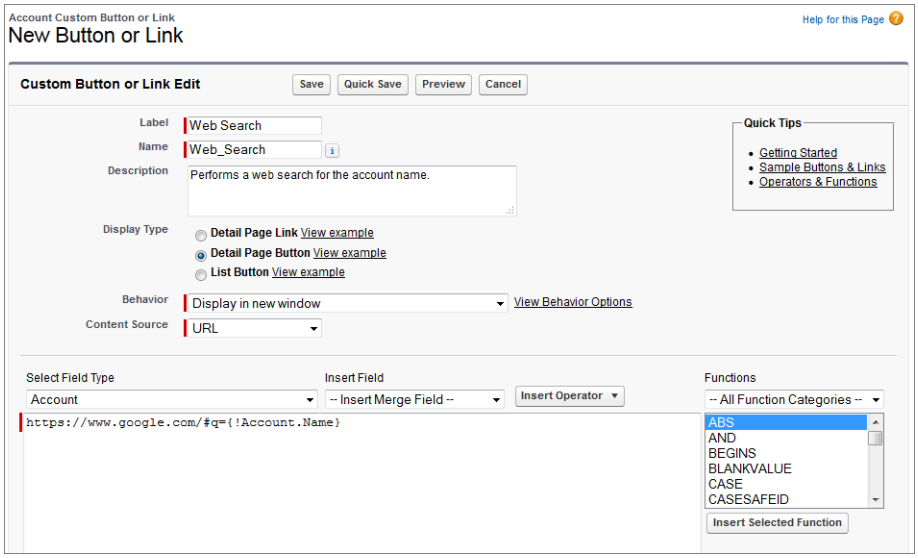
If you want the button or link to launch a custom page or other code, consider a Visualforce page.

1. From the management settings for the object that you want to edit, go to Buttons, Links, and Actions.

   Note: Custom buttons aren't available on the User object or custom home pages. Custom buttons and links are available for activities under the individual object management settings for tasks and events. To override a standard button that applies to both tasks and events, go to the object management settings for activities.

2. Click **New Button or Link**. Alternatively, click **Default Custom Links** to add a predefined custom link.
3. Enter the button or link attributes.

   Here's an example of the attributes for a button that performs a web search for an account's name.



4. To validate all Salesforce merge fields and functions, click **Check Syntax**.
5. Click **Save** when you're finished, or click **Quick Save** to save and continue editing. Saving validates the URL you defined if you set the content source to URL.
6. To open a button or link using settings other than the user's default browser settings, click **Window Open Properties** on the button or link's detail page.
7. To view all references to the new button or link, click **Where is this used?** on its detail page.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

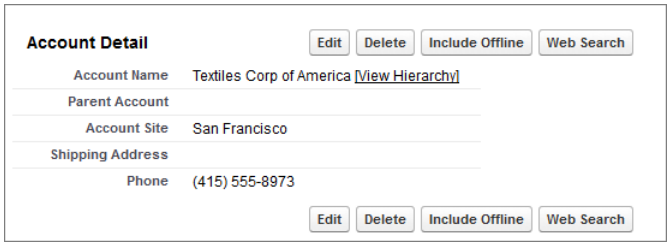
To create or change custom buttons or links:

- "Customize Application"

Custom links for users are automatically added to the Custom Links section of the user detail page. You can add page buttons only to the Button section of a page layout.

Note: A link URL can be up to 2,048 bytes. When data is substituted for the tokens in the URL, the link can exceed 3,000 bytes. Some browsers enforce limits on the maximum URL length.

Before you can use your custom buttons and links, add them to an object's page layout.You can then see and use the button or link on a record detail page.



IN THIS SECTION:

Custom Button and Link Fields

This table defines the fields that are available when you create a custom button or link.

Constructing Effective Custom Links

Use these methods to construct effective custom links.

Editing Window Open Properties

Merge Fields for Custom Buttons and Links

Adding Default Custom Links

Custom Link Best Practices

Custom buttons and links can streamline actions within Salesforce, or integrate Salesforce data with external URLs, applications, or systems. Use these best practices to get the most out of your custom links.

SEE ALSO:

Find Object Management Settings

Custom Button and Link Considerations

Custom Button and Link Samples

Custom Button and Link Fields

This table defines the fields that are available when you create a custom button or link.

Attribute Name	Description
`Label`	Enter the text that displays in the user interface for the custom button or link.
`Name`	Accept or enter the unique name to use for the button or link when it's referenced from a merge field. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Namespace Prefix`	Enter the prefix that uniquely identifies your package. In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc are not recognized as unique. Your namespace prefix must be globally unique across all Salesforce organizations. It keeps your managed package under your control exclusively.
`Protected Component`	Optionally limit use in a subscriber org. Protected components can't be linked to or referenced by components created in a subscriber org. A developer can delete a protected component in a future release without worrying about failing installations. However, once a component is marked as unprotected and is released globally, the developer can't delete it.
`Description`	Enter text that distinguishes the button or link from others. This text displays only to administrators when setting up buttons and links.
`Display Type`	Determine where the button or link appears on page layouts. **Detail Page Link** Adds the link to the Custom Links section of your page layouts. **Detail Page Button** Adds the custom button to a record's detail page. You can add detail page buttons to the Button section of a page layout. **List Button** Adds the custom button to a list view, search result layout, or related list. You can add list buttons to the Related List section of a page layout or the List View and Search Result layouts. For list buttons, Salesforce selects the **Display Checkboxes (for Multi-Record Selection)** option to include a checkbox next to each record in the list. Users can select the records they want applied to the action on the list button. If your custom button doesn't require the user to select records, deselect this option.

EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions except **Database.com**

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Attribute Name	Description
`Behavior`	Choose the outcome of clicking the button or link. When applicable, some settings have default values. For example, if you choose **Display in new window**, the default height of a new window is 600 pixels. See Editing Window Open Properties.
`Content Source`	Choose whether to use a URL, s-control, JavaScript action, or Visualforce page as the content of the button or link. Salesforce checks the correctness of URLs in custom links and custom buttons. When you create or edit custom links or buttons that contain invalid URL markup, such as scripts, Salesforce blocks the links from rendering. Instead, an error message is shown on hover. Reconfigure the URLs to be valid and well formed. The URL can be a relative URL or an absolute `http://`, `https://`, `file://`, `ftp://`, or `mailto://` address. Invalid URLs in custom links or custom buttons created before Spring '13 aren't checked for correctness until you edit them.

`Content`	Enter the content of the button or link for buttons and links of type URL or OnClick JavaScript.
	• To insert a field, choose the field type from Select Field Type and choose a field from Insert Field.
	• To insert activity merge fields, select *`Event`* or *`Task`* from `Select Field Type`.
	• To insert an operator, choose the appropriate operator icon from the `Insert Operator` drop-down list.
	• To insert a function, double-click its name in the list, or select it and click **Insert Selected Function**. To filter the list of functions, choose a category from the `Functions` drop-down list. Select a function and click **Help on this function** to view a description and examples of formulas using that function.
	Tip: Internet standards require special encoding for URLs. Salesforce encodes the text from any merge field you insert into a link. Encode additional text in your link manually. For example, to generate the following URL:
	`http://www.google.com/search?q={!user.name} Steve Mark 50%`
	Use this content:
	`http://www.google.com/search?q={!user.name}+Steve+Mark+50%25`
	Salesforce strips double quotes from URLs when the content source is a URL. If you must use double quotes, encode them manually. For example, to generate the URL `http://www.google.com/search?q="salesforce+foundation"`, use this content:
	`http://www.google.com/search?q=%22salesforce+foundation%22`

`Link Encoding`	Choose the encoding setting. Encoding defaults to Unicode (UTF-8). Change the default encoding setting if the target of a link requires data in a different format. Encoding is available if your Content Source is URL.

Constructing Effective Custom Links

Use these methods to construct effective custom links.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Use merge fields to pass Salesforce data

You can add merge fields to a custom link to pass data from your Salesforce records, user information, or company information to another website or application.

For example, you can create custom links that search Google for an account name or look up an address on a map.

```
http://google.com/search?q={!Account.Name}
```

```
http://maps.google.com?q={!Account.ShippingStreet} {!Account.ShippingCity}
{!Account.ShippingState} {!Account.ShippingPostalCode}
```

Note: Custom links don't support data type conversion. When creating custom links to pass data from one Salesforce field to another, the data must match the data type of the fields in which you are placing it. For example, if you have numeric data, you must pass it to a numeric field.

Substitute symbols for certain characters or use `URLENCODE()`

Due to URL encoding standards set forth by the World Wide Web Consortium (W3C), certain "unsafe" characters, such as spaces and punctuation marks, can't be passed through a URL. Custom buttons and links escape these characters, so you don't have to URL-encode them.

If you do need to encode your URL, use the `URLENCODE()` function in the merge field like so: `{!URLENCODE(text)}` and replace *`text`* with the merge field or text string that you want to encode.

For example: If the merge field `foo__c` contains `<B>Mark's page<b>`, `{!URLENCODE(foo_c)}` results in: `%3CB%3EMark%27s%20page%3C%2Fb%3E`.

Use `URLFOR()` to link to Visualforce pages

To link to a Visualforce page, use `URLFOR()` with the relative path to the page, which is `/apex/PageName`. For example, to link to a Visualforce page called MissionList that isn't associated with a record, use the following syntax.

```
{! URLFOR( "/apex/MissionList" ) }
```

When you use `URLFOR()` with a Visualforce page, and you want to pass a record ID into the page, you must pass the ID in as a parameter.

```
{! URLFOR( "/apex/Mission", null, [id=Mission__c.Id] ) }
```

Use the `$Action` global variable and `URLFOR()` to link to Salesforce pages

When creating a custom button or link that points to a page in Salesforce, use the `$Action` global variable to construct the link, instead of pasting in the path to the page. Then, if your organization is moved to another server or the URL to the page changes, the link still points to the right place.

To construct the link, use the `URLFOR()` formula function with the `$Action` variable.

```
{!URLFOR( $Action.Case.NewCase, Account.Id )}
```

This custom link on the Account object opens the New Case form, creating the case as a child of the account record. You can use this process for any object that has a lookup to the Account object. To create a record that is not a child of another record, or if two objects have no relationship, you can use `$ObjectType.ObjectName` as the second argument. For example:

```
{!URLFOR( $Action.Case.NewCase, $ObjectType.Case )}
```

`$Action` global variables expect either a record ID or the `$ObjectType`. For example, the next two formulas create links to the tab and detail page for an account, respectively.

```
{!URLFOR( $Action.Account.Tab, $ObjectType.Account )}
```

```
{!URLFOR( $Action.Account.View, Some_Account_Lookup__c.Id )}
```

The `URLFOR()` function takes additional optional arguments that get passed into the destination as query string parameters. You can use these arguments when overriding a standard action with a Visualforce page to pass in the additional parameters needed by the Visualforce page or its controller. For example, if when closing a case you want to change the value of a custom field on the case called Actual Delivery Date to today, you could use:

```
{!URLFOR($Action.Case.CloseCase, Case.Id, [ actualDeliveryDate=TODAY()] )}
```

You can then override the Close Case action with a Visualforce page and handle setting the value of the Actual Delivery Date field either in that Visualforce page or its controller. See *Using Query String Parameters in a Visualforce Page* for more information.

SEE ALSO:

Custom Link Example: Link to Documents

Custom Link Example: Link to Files in Chatter

Custom Link Example: Link to Reports

Editing Window Open Properties

Custom buttons and links can open in different types of windows. If you have selected a custom button or link to open in a popup window, set the window properties. If you leave the window properties blank, the custom button or link will use the default settings of the user's browser.

To edit the window open properties:

1. From the management settings for the object whose button or link you want to edit, go to Buttons, Links, and Actions, and then click the button or link name. Custom buttons are not available on the user object.
2. Click **Window Open Properties**.
3. Edit the following properties.

Window Property	Description
`Width`	The width (in pixels) of the popup.
`Height`	The height (in pixels) of the popup.
`Window Position`	The location on the screen where the popup should open.

EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

S-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit custom button or link properties:

- "Customize Application"

Window Property	Description
`Resizeable`	Allow users to resize the popup window.
`Show Address Bar`	Show the browser's address bar which contains the URL.
`Show Scrollbars`	Show browser scrollbars for the popup.
`Show Toolbars`	Show the browser toolbars. Toolbars normally contain navigation buttons like Back, Forward, and Print.
`Show Menu Bar`	Show the browser menus. The menus typically contain option like **File** and **Edit**.
`Show Status Bar`	Show the status bar at the bottom of the browser.

Note: Some properties may not be available depending on the behavior of the custom button or link. For example, if you chose *`Execute JavaScript`*, no window open properties are available.

4. Save your changes.

SEE ALSO:

Define Custom Buttons and Links

Merge Fields for Custom Buttons and Links

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions except **Database.com**

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Syntax and Formatting

When you insert a merge field in a custom button or link, the syntax consists of an open curly brace and exclamation point, followed by the object name, a period, the field name and a closing curly brace.

```
{!Object_Name.Field_Name}
```

To ensure that you're using the correct syntax, select merge fields from the drop-down list in the editor for custom buttons and links.

Tips

- To insert activity merge fields, select **Event** or **Task** from the Select Field Type drop-down list.
- You can add links quickly to the sidebar by using the standard home page's Custom Links component.

Warning: The standard home page's Custom Links component doesn't support

- Merge fields
- Functions, such as `URLFOR`
- JavaScript execution
- Customizable window opening properties

SEE ALSO:

Merge Fields Overview

Custom Button and Link Considerations

Adding Default Custom Links

1. From the management settings for the appropriate object, go to Buttons, Links, and Actions or to Buttons and Links.
2. Click **Default Custom Links**.
3. Next to a sample link you want to add, click **Add Now!**.
4. Change the default data for the link, as necessary.
5. Choose **Save**.
6. Edit the page layout for the appropriate tab to display the new link.

SEE ALSO:

Define Custom Buttons and Links

EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

USER PERMISSIONS

To create or change custom links:

- "Customize Application"

Custom Link Best Practices

Custom buttons and links can streamline actions within Salesforce, or integrate Salesforce data with external URLs, applications, or systems. Use these best practices to get the most out of your custom links.

Note: Salesforce checks the correctness of URLs in custom links and custom buttons. When you create or edit custom links or buttons that contain invalid URL markup, such as scripts, Salesforce blocks the links from rendering. Instead, an error message is shown on hover. Reconfigure the URLs to be valid and well formed. The URL can be a relative URL or an absolute `http://`, `https://`, `file://`, `ftp://`, or `mailto://` address. Invalid URLs in custom links or custom buttons created before Spring '13 aren't checked for correctness until you edit them.

Best Practices for Advanced Users

Salesforce has no plans to change field names; however, that doesn't guarantee that field names won't change in the future. Therefore, custom links that include Salesforce fields may change how they are mapped.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Pass Session IDs with SSL

Never pass session IDs to an "`http`" URL. Instead, pass session IDs with a secure sockets layer (SSL) "`https`" URL. Any data that is passed to other applications hosted on the Internet should always use SSL since the URL may contain sensitive customer information.

Single Sign-On

Use custom links to pass a session ID to support Single Sign-On (SSO), so users can avoid multiple logins to web applications that your organization hosts to manage Salesforce data. Construct your custom link to pass the `{!User_Session_ID}` merge field, which allows users to access all authorized resources during a single authentication. External systems can access Salesforce resources using a web service, which allows organizations to communicate data without intimate knowledge of each other's IT systems behind a firewall.

Send the Server Date in the URL

Some integration projects need custom links to include a server date to know the Salesforce initiation date. The Salesforce server date can be passed to external systems using custom links. For example, `http://someurl.com/somepath?current_date={!Today}`. Dates use the Pacific time zone.

Avoid Double Sets of Tabs in a Browser

Unlimited Edition and Enterprise Edition users can build a custom link to perform an action that keeps users in the same browser window and doesn't display a double set of tabs. Create a Visualforce page that contains the following code, replacing `<REGULAR_WIL>` with your regular or existing custom link.

```
<script language="JavaScript">
function redirect() {
parent.frames.location.replace("<REGULAR_WIL>")
}
redirect();
</script>
```

Override Standard Buttons and Tab Home Pages

You can override the behavior of standard buttons on record detail pages. You can also override the tab home page that displays when a user clicks a standard, custom, or external object tab.

Button overrides are global. For example, if you override the New button on opportunities, your replacement action takes effect wherever that action is available, such as:

- Opportunities home page
- Any opportunities related lists on other objects, such as accounts
- Create New drop-down list in the sidebar
- Any browser bookmarks for this Salesforce page

Important: Before you override a standard button, review the considerations for overriding standard buttons.

1. From the object management settings for the object you want to set an override for, go to Buttons, Links, and Actions.
2. Click **Edit** next to the button or tab home page you want to override.
3. Pick the type of override you want associated with the action.
   - **No Override (use default)**—Use a custom override provided by an installed package. If there isn't one installed, the standard Salesforce behavior is used.
   - **Standard Salesforce Page**—This option is available only for subscribers who are overriding the actions on an installed custom object. If selected, the standard Salesforce behavior is used.
   - **Custom S-Control**—Use the behavior from an s-control.

     Important: Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected, and can still be edited.

   - **Visualforce Page**—Use the behavior from a Visualforce page.
4. Select the name of the s-control or Visualforce page you want to run when users click the button or tab.

   When overriding buttons with a Visualforce page, you can select only Visualforce pages that use the standard controller for the object on which the button appears. For example, if you want to use a page to override the **Edit** button on accounts, the page markup must include the `standardController="Account"` attribute on the `<apex:page>` tag:

   ```
   <apex:page standardController="Account">

       ... page content here ...

   </apex:page>
   ```

   When overriding tabs with a Visualforce page, you can select only Visualforce pages that use the standard list controller for that tab, pages with a custom controller, or pages with no controller.

   When overriding lists with a Visualforce page, you can select only Visualforce pages that use a standard list controller.

   When overriding the **New** button with a Visualforce page, you have the option to skip the record type selection page. If you choose this option, new records you create aren't forwarded to the record type selection page. Salesforce assumes that your Visualforce page is already handling record types.
5. Optionally, enter any comments to note the reason for this change.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce overrides also available in: **Contact Manager**, **Group**, and **Professional** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To override standard buttons and tab home pages:
- "Customize Application"

To reset button and tab home page overrides:
- "Customize Application"

6. Click **Save**.

Note: A standard button (New, Edit, View, Delete, and Clone) overridden with a Visualforce page doesn't show up in Salesforce1 unless the Visualforce page is enabled for Salesforce mobile apps. Overriding standard list and tab controls isn't supported in Salesforce1.

IN THIS SECTION:

Considerations for Overriding Standard Buttons
Before you override a standard button, review these considerations.

Remove Overrides for Standard Buttons and Tab Home Pages

SEE ALSO:

Remove Overrides for Standard Buttons and Tab Home Pages

Considerations for Overriding Standard Buttons

Before you override a standard button, review these considerations.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce overrides also available in: **Contact Manager**, **Group**, and **Professional** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To override standard buttons:
- "Customize Application"

Implementation Tips

- If you override a standard button in Salesforce, that button is still available in Connect Offline or Salesforce Classic Mobile, but it retains its original behavior.
- A standard button (New, Edit, View, Delete, and Clone) overridden with a Visualforce page doesn't show up in Salesforce1 unless the Visualforce page is enabled for Salesforce mobile apps. Overriding standard list and tab controls isn't supported in Salesforce1.
- The View standard button refers to all links in Salesforce that display the detail page for a record. Overriding the View standard button reroutes all these links.
- You can have both a Visualforce or s-control override and a Lightning Page override set for the View action at the same time on the same object. In Lightning Experience, the Lightning Page override displays. In Salesforce Classic, the Visualforce or s-control override displays.
- If you have the View action overridden with a Visualforce page in Salesforce Classic, the Setup menu on that object record page in Lightning Experience displays the Edit Page option. Selecting **Edit Page** in Lightning Experience on an object page that's overridden with a Visualforce page in Salesforce Classic lets you create a custom Lightning Experience record page for that object in the Lightning App Builder.
- If a button isn't available for overrides, you can still hide it on the page layout.
- Button overrides affect everywhere that action or behavior is available. For example, overriding the New button on an account also overrides the account option in the Create New drop-down list in the sidebar.
- Person Account records use any standard button overrides you make for accounts. Person Account records also use any overrides for the View Self-Service and Enable Self-Service buttons you make for contacts.
- If your organization uses the Console tab, overrides for the Edit and View buttons for an object don't affect the Edit and View buttons in the mini page layouts. Pages that display due to overrides display in the console without the header or sidebar.
- To replace a standard button with a custom button, first define the custom button, then customize the page layout to hide the standard button and display the custom one in its place
- Because events and tasks don't have their own tabs, you can only override their standard buttons and links.

- When you override the Edit button with a Visualforce page, you're also overriding default logic that checks whether the record has been locked for approval. However, you can perform the same check by implementing the `Approval.lock` Apex method.

Limitations

- You can override buttons on the detail page but not the edit page of a record.
- You can only override these standard buttons: New, View, Edit, and Delete.
- You can't add button overrides to an AppExchange package.
- You can't change buttons on lookup dialogs, reports, or tabs. However, you can change the buttons on list view and search result layouts under search layouts.
- You can't relabel or relocate standard buttons on a record detail page.

SEE ALSO:

Override Standard Buttons and Tab Home Pages

Remove Overrides for Standard Buttons and Tab Home Pages

1. From the management settings for the object whose button you want to edit, go to Buttons, Links, and Actions.
2. Click **Edit** next to the overridden button or tab link.
3. Select **No Override (default behavior)**.
4. Click **OK**.

EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Visualforce overrides also available in: **Contact Manager**, **Group**, and **Professional** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To override standard buttons and tab home pages:
- "Customize Application"

To reset button and tab home page overrides:
- "Customize Application"

Custom Button and Link Samples

Use samples of custom Salesforce buttons and links to determine whether they can work for you.

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Custom Link Example: Link to Documents

Use custom links to reference documents from a Salesforce record detail page.

Custom Link Example: Link to Files in Chatter

Use custom links to reference files from Chatter.

Custom Link Example: Link to Reports

Use custom links to run reports with filtered results from a Salesforce record detail page. For example, let's say you frequently run a mailing list report for the contacts related to an account. You can create a custom link for accounts that links directly to a report that is automatically filtered to the account you are viewing. In this case, your custom link must pass the account's unique record ID to the report.

Displaying Alerts

This example creates a button that opens a popup dialog with a welcome message containing the user's first name.

Getting Record IDs

This example creates a button that opens a popup window listing record IDs for user selected records. This is useful when testing to ensure you have the correct record IDs before processing them further.

Create a Custom Button for Performing Mass Deletes

This example creates a button that can be added to activity-related lists and list views and allows users to delete selected records at the same time.

Passing Record IDs to an External System

You can use Salesforce record IDs as unique identifiers for integrating with an external system. This example creates a button that calls a Visualforce page to determine the record IDs of selected records and passes them in a URL query parameter to an external Web page called "`www.yourwebsitehere.com`."

Reopening Cases

This example creates a button that can be added to cases related lists so users can reopen several cases on an opportunity at once.

International Maps

This example creates a link that displays a country-specific Google map.

SEE ALSO:

Define Custom Buttons and Links

Custom Link Example: Link to Documents

Use custom links to reference documents from a Salesforce record detail page.

1. Create a folder on the Documents tab to which all users have access.
2. Upload the document to that folder.
3. From the Documents tab, choose the folder and click **Go**.
4. Click **View** next to the document.
5. Copy the document's URL from the browser. For example,

   `https://na1.salesforce.com/servlet/servlet.FileDownload?file=015300000000xvU`.
6. Use everything after the domain portion of the URL to create your custom link. Using the example in the previous step, your link would point to `/servlet/servlet.FileDownload?file=015300000000xvU`.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

SEE ALSO:

Constructing Effective Custom Links
Custom Link Example: Link to Files in Chatter
Custom Link Example: Link to Reports

Custom Link Example: Link to Files in Chatter

Use custom links to reference files from Chatter.

1. Upload a file to the Files tab.
2. When the upload is finished, from the Upload dialog box, click **Share settings**.
3. Click **Anyone with link**.
4. Copy the document's URL from the Share via Link dialog box. For example,

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

```
https://na1.salesforce.com/sfc/p/D0000000JsES/a/D000000001dd/aiq8UPJ5q5i6Fs4Sz.IQLKUERsWYdbAm320cjqWnkfk=.
```

5. Use everything after the domain portion of the URL to create your custom link. Using the example in the previous step, your link would point to `/sfc/p/D0000000JsES/a/D000000001dd/aiq8UPJ5q5i6Fs4Sz.IQLKUERsWYdbAm320cjqWnkfk=`.

SEE ALSO:

Constructing Effective Custom Links
Custom Link Example: Link to Documents
Custom Link Example: Link to Reports

Custom Link Example: Link to Reports

Use custom links to run reports with filtered results from a Salesforce record detail page. For example, let's say you frequently run a mailing list report for the contacts related to an account. You can create a custom link for accounts that links directly to a report that is automatically filtered to the account you are viewing. In this case, your custom link must pass the account's unique record ID to the report.

1. Copy the ID for the type of record by which you want to filter your report (in this example, an account). To do so, view the record and copy the 15-character ID from the last part of the URL. For example, `https://na1.salesforce.com/001200030012j3J`.
2. From the Reports tab, create the report you want by either customizing a standard report or creating a custom report.
3. Filter the report by the record ID you copied. For example, "Account ID equals 001200030012j3J".
4. Run the report to verify that it contains the data you expect.
5. Click **Customize**.
6. Click **Save** or **Save As** to save the report to a public folder accessible by the appropriate users. **Save** doesn't create a custom report, whereas **Save As** does.
7. Run the report and copy the report's URL from the browser.
8. Begin creating your custom link. Set the `Content Source` field to URL. In the large formula text area, paste the report URL you copied. Remember to omit the domain portion `https://na1.salesforce.com`.
9. Add the custom link to the appropriate page layouts.
10. Verify that the new custom link works correctly.

Tip: When creating a report for use in a custom link, set date ranges and report options generically so that report results include data that can be useful for multiple users. For example, if you set a date range using the "Created Date" of a record, set the Start Date far enough in the past to not exclude any relevant records and leave the End Date blank. If you scope the report to just "My" records, the report might not include all records that a user can see. Try setting the report options to "All visible" records.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

SEE ALSO:

Constructing Effective Custom Links
Custom Link Example: Link to Documents
Custom Link Example: Link to Files in Chatter

Displaying Alerts

This example creates a button that opens a popup dialog with a welcome message containing the user's first name.

1. Define a button with the following attributes.

- `Display Type` Detail Page Button
- `Behavior` Execute JavaScript
- `Content Source` OnClick JavaScript
- Use the following sample code.

```
alert ("Hello {!$User.FirstName}");
```

2. Add the button to the appropriate page layout.

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Custom Button and Link Samples

Getting Record IDs

This example creates a button that opens a popup window listing record IDs for user selected records. This is useful when testing to ensure you have the correct record IDs before processing them further.

1. Define a button with the following attributes.
   - `Display Type` List Button

     Note: Select **Display Checkboxes (for Multi-Record Selection)** so users can select multiple records in the list before clicking the button.
   - `Behavior` Execute JavaScript
   - `Content Source` OnClick JavaScript
   - Use the following sample code.

```
idArray = {!GETRECORDIDS($ObjectType.Contact)};
alert("The Ids you have selected are: "+idArray);
```

   Note: This example is for contacts. Change the object type for a different type of record.

2. Add the button to the appropriate related list on a page layout or list view layout.

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

Custom Button and Link Samples

Create a Custom Button for Performing Mass Deletes

This example creates a button that can be added to activity-related lists and list views and allows users to delete selected records at the same time.

1. Define a button for events with the following attributes.
   - `Display Type` List Button

     Note: Select **Display Checkboxes (for Multi-Record Selection)** so users can select multiple records in the list before clicking the button.

   - `Behavior` Execute JavaScript
   - `Content Source` OnClick JavaScript
   - Use the following sample code.

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
{!REQUIRESCRIPT("/soap/ajax/9.0/connection.js")}

var records = {!GETRECORDIDS( $ObjectType.Event )};
var taskRecords = {!GETRECORDIDS( $ObjectType.Task)};
records = records.concat(taskRecords);


if (records[0] == null) {
alert("Please select at least one record.") }
else {

var errors = [];
var result = sforce.connection.deleteIds(records);
if (result && result.length){
var numFailed = 0;
var numSucceeded = 0;
for (var i = 0; i < result.length; i++){
var res = result[i];
if (res && res.success == 'true'){
numSucceeded++;
} else {
var es = res.getArray("errors");
if (es.length > 0) {
errors.push(es[0].message);
}
numFailed++;
}
}
```

```
if (numFailed > 0){
alert("Failed: " + numFailed + "\nSucceeded: " + numSucceeded + " \n Due to: " +
errors.join("\n"));
} else {
alert("Number of records deleted: " + numSucceeded);
}
}
window.location.reload();
}
```

2. Add the button to your activity list views.
3. Add the button to any page layout that contains an activity-related list. The button deletes any selected task or event in the list.

You can install custom buttons from the Mass Delete app at http://sites.force.com/appexchange.

SEE ALSO:

Custom Button and Link Samples

Passing Record IDs to an External System

You can use Salesforce record IDs as unique identifiers for integrating with an external system. This example creates a button that calls a Visualforce page to determine the record IDs of selected records and passes them in a URL query parameter to an external Web page called "`www.yourwebsitehere.com`."

1. Create a Visualforce page that uses the GETRECORDIDS function to retrieve a list of selected records:

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
<script type="text/javascript">

idArray = {!GETRECORDIDS($ObjectType.Account)};

window.location.href="http://www.yourwebsitehere.com?array="+idArray;

</script>
```

Note: Replace `www.yourwebsitehere.com` with your own URL.

2. Define a button for accounts with the following attributes.
   - `Display Type` List Button

     Note: Select **Display Checkboxes (for Multi-Record Selection)** so users can select multiple records in the list before clicking the button.

   - `Behavior` Display in existing window with sidebar
   - `Content Source` Visualforce Page
   - Select the Visualforce page you created in the first step.
3. Add the button to the appropriate page layout or list view layout.

SEE ALSO:

Custom Button and Link Samples

Reopening Cases

This example creates a button that can be added to cases related lists so users can reopen several cases on an opportunity at once.

1. Define a button for cases with the following attributes.
   - `Display Type` List Button

     Note: Select **Display Checkboxes (for Multi-Record Selection)** so users can select multiple records in the list before clicking the button.

   - `Behavior` Execute JavaScript
   - `Content Source` OnClick JavaScript
   - Use the following sample code.

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
{!REQUIRESCRIPT ("/soap/ajax/13.0/connection.js")} var records =
                  {!GETRECORDIDS($ObjectType.Sample)}; var newRecords = []; if
(records[0] == null) { alert("Please select at least one row") } else {
                  for (var n=0; n<records.length; n++) { var c = new
sforce.SObject("Case"); c.id = records[n]; c.Status = "New";
               newRecords.push(c); } result = sforce.connection.update(newRecords);
 window.location.reload(); }
```

Note: This example references the AJAX Toolkit, which is available if API access is enabled. See https://developer.salesforce.com/page/Integration.

2. Add the button to your opportunity page layouts by editing the Cases related list.

Tip: Use this button on any page layout that contains the cases related list, such as account or contact page layouts.

Note: Notice the check for `records[0] == null`, which displays a message to users when they don't select at least one record in the list.

SEE ALSO:

Custom Button and Link Samples

International Maps

This example creates a link that displays a country-specific Google map.

1. Define a link for accounts with the following attributes.
   - `Display Type` Detail Page Link
   - `Behavior` Display in new window
   - `Content Source` URL
   - Use the following sample code.

USER PERMISSIONS

To create or change custom buttons or links:
- "Customize Application"

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

```
{!
IF(Sample.BillingCountry = "US",
"http://maps.google.com/maps?q="&Sample.BillingStreet&
"+"&Sample.BillingCity&"+"&Sample.BillingState&"+"&Sample.BillingCountry,
(IF(Sample.BillingCountry = "UK",
"http://maps.google.co.uk/maps?q="&Sample.BillingStreet
&"+"&Sample.BillingCity&"+"&Sample.BillingCountry,
"http://maps.google.com")))
}
```

2. Add the link to your account page layout.

SEE ALSO:

Custom Button and Link Samples

Custom Button and Link Considerations

Keep these considerations in mind when working with custom buttons and links.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

Implementation Tips

- Custom buttons display at the top and bottom of the detail page to the right of all standard buttons.
- Custom buttons aren't distinguished from standard buttons in any graphical way. However, you can recognize them by their location on the right of all standard buttons.
- If the button bar gets too wide on the detail page layout, the browser displays a horizontal scroll bar. If the button bar gets too wide on the list view, search result, tagging result, or related list layouts, the buttons wrap.
- Custom buttons are available for activities under the individual setup links for tasks and events. To add a custom button to an activity list view or search layout, first create a custom list button in tasks or events. Next, add it to your activity list view or search result layouts. You can override a button that applies to both tasks and events.
- Person Account records use the custom buttons and links you have made for accounts.
- If your organization uses the Console tab, list buttons are available in `Mass Action`. List buttons won't display in the mini page layouts. Pages that display due to custom buttons and links display in the console without the header or sidebar.
- If you get an error message when overriding a button that appears in a list, try calling the s-control using the URLFOR function.
- When creating custom buttons, be aware of any validation rules your organization has for records on that object. For example, a custom list button that changes case status may conflict with a case validation rule. In this scenario, Salesforce displays the error message for the validation rule when users click the custom button.
- To replace a standard button with a custom button, first define the custom button, then customize the page layout to hide the standard button and display the custom one in its place
- Visualforce pages used as custom links on the home page can't specify a controller.
- Visualforce pages used as custom buttons or links on detail pages must specify a standard controller of the same object.
- Visualforce pages used as custom list buttons must use a standard list controller of the same object.

Best Practices

- Use formula functions in custom buttons with caution. Because functions run on the server before your HTML or JavaScript is passed to the browser, they can only evaluate information that exists at that time. Avoid using functions like IF to evaluate conditions that only exist when the code reaches the browser, such as the value of a JavaScript variable that is returned from your code.
- To prevent a user from performing a particular action, such as creating or editing, change the user's permissions rather than hiding the standard button. Hiding a standard button removes it from a page layout but the link is still available and users can navigate to the new or edit page manually.

- Use global variables to access special merge fields for components like custom buttons, links, and s-controls. For example, the `$Request` global variable allows you to access query parameters inside a snippet, s-control, or custom button.
- Don't select `Display Checkboxes (for Multi-Record Selection)` for list buttons that link to a URL that doesn't support post operations. Checkboxes display next to records in a list if you have selected `Display Checkboxes (for Multi-Record Selection)` for at least one list button in that list. However, selected records in this list are unaffected when clicking a custom list button that doesn't have this option selected.

Limitations

- A custom link's label can't exceed 1,024 characters.
- A link URL can be up to 2,048 bytes. When data is substituted for the tokens in the URL, the link can exceed 3,000 bytes. Some browsers enforce limits on the maximum URL length.
- Custom buttons that call JavaScript aren't supported in Lightning Experience.
- Custom buttons with a content source of OnClick JavaScript aren't supported in the Salesforce1 mobile app.
- If you use URL custom buttons to pass parameters to standard pages in Salesforce Classic—such as pre-populating fields when creating a record—this behavior doesn't work in Salesforce1 or Lightning Experience.
- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons and links that call JavaScript aren't supported.
- Custom buttons aren't available for Web-to-Lead, Web-to-Case, the Case Teams related list, or the user object.

SEE ALSO:

Define Custom Buttons and Links

Custom Button and Link Samples

Viewing References to Salesforce Components

You can view a list of all the areas in Salesforce that reference a component. For example, view the custom links, custom buttons, or page layouts that reference another component, such as a Visualforce page or static resource. To do this, click **Where is this used?** from the detail page of the component. Salesforce lists the type of component that references the component and the label for that component. Click any item in the list to view it directly.

SEE ALSO:

Define Custom Buttons and Links

EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions

Visualforce pages and custom components available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or change custom buttons or links:

- "Customize Application"

To create, edit, and delete Visualforce pages and custom components:

- "Customize Application"

To clone, edit, or delete static resources:

- "Customize Application"

External Data Integration

Sync Data Between Salesforce and Heroku Postgres with Heroku Connect

Heroku Connect lets you sync data between Salesforce and Heroku Postgres.

Using Heroku Connect with Heroku Postgres, you can build Heroku apps that interact with your Salesforce data using your favorite language, like Ruby, Node.js, Python, PHP, Java, Scala or Clojure or web framework like Rails, Django, or Play. For more information, refer to the `Heroku Connect website` and the `Heroku Connect documentation` on the Heroku Dev Center website.

EDITIONS

Available in: Salesforce Classic

Available for: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Access External Data with Salesforce Connect

Salesforce Connect

Salesforce Connect

Salesforce Connect provides seamless integration of data across system boundaries by letting your users view, search, and modify data that's stored outside your Salesforce org. For example, perhaps you have data that's stored on premises in an enterprise resource planning (ERP) system. Instead of copying the data into your org, you can use external objects to access the data in real time via web service callouts.

Traditionally, we've recommended importing or copying data into your Salesforce org to let your users access that data. For example, extract, transform, and load (ETL) tools can integrate third-party systems with Salesforce. However, doing so copies data into your org that you don't need or that quickly becomes stale.

In contrast, Salesforce Connect maps Salesforce *external objects* to data tables in external systems. Instead of copying the data into your org, Salesforce Connect accesses the data on demand and in real time. The data is never stale, and we access only what you need. We recommend that you use Salesforce Connect when:

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- You have a large amount of data that you don't want to copy into your Salesforce org.
- You need small amounts of data at any one time.
- You want real-time access to the latest data.

Even though the data is stored outside your org, Salesforce Connect provides seamless integration with the Force.com platform. External objects are available to Salesforce tools, such as global search, lookup relationships, record feeds, and the Salesforce1 app. External objects are also available to Apex, SOSL,SOQL queries, Salesforce APIs, and deployment via the Metadata API, change sets, and packages.

For example, suppose that you store product order information in a back-office ERP system. You want to view those orders as a related list on each customer record in your Salesforce org. Salesforce Connect enables you to set up a lookup relationship between the customer object (parent) and the external object (child) for orders. Then you can set up the page layouts for the parent object to include a related list that displays child records.

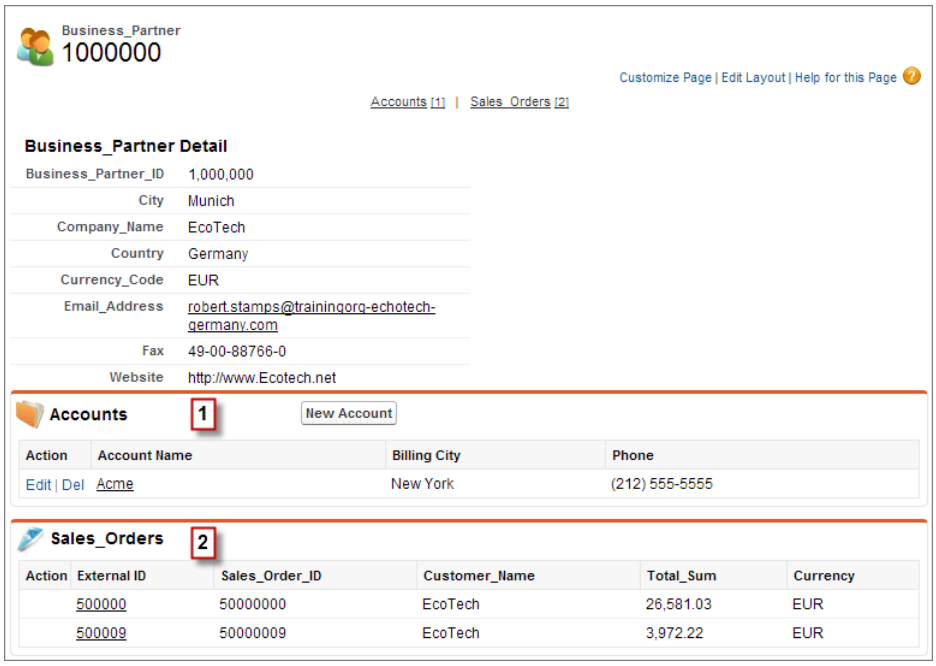
Going a step further, you can update the orders directly from the related list on the customer record. By default, records that are based on external objects are read-only. Making an external data source writable lets users create, modify, or delete records that are based on external objects from that data source.

Note: You can make external objects writable only when an OData or Apex custom adapter is used to connect your Salesforce org to the external data. Writable external objects aren't supported with the cross-org adapter.

For information about using Apex DML write operations on external object records, see the *Force.com Apex Code Developer's Guide*.

Example: This screenshot shows how Salesforce Connect can provide a seamless view of data across system boundaries. A record detail page for the Business_Partner external object includes two related lists of child objects. The external lookup relationships and page layouts enable users to view related data from inside and from outside the Salesforce org on a single page.

- Account standard object (1)
- Sales_Order external object (2)



IN THIS SECTION:

Salesforce Connect Adapters

Salesforce Connect uses a protocol-specific adapter to connect to an external system and access its data. When you define an external data source in your organization, you specify the adapter in the `Type` field.

Identity Type for External Data Sources

On the external data source, the `Identity Type` field specifies whether your organization uses one set or multiple sets of credentials to access the external system. Each set of credentials corresponds to a login account on the external system.

Record IDs for Salesforce Connect External Objects

The first time a data row is retrieved from an external system, the external object record is assigned a Salesforce ID. Each record ID remains associated with the same external data row, unless the external object is deleted from the org.

SEE ALSO:

External Object Relationships

Considerations for Salesforce Connect—All Adapters

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

Access External Data with a Custom Adapter for Salesforce Connect

Salesforce Connect Adapters

Salesforce Connect uses a protocol-specific adapter to connect to an external system and access its data. When you define an external data source in your organization, you specify the adapter in the `Type` field.

These adapters are available for Salesforce Connect.

Salesforce Connect Adapter	Description	Where to Find Callout Limits
Cross-org	Uses the Force.com REST API to access data that's stored in other Salesforce orgs.	No callout limits. However, each callout counts toward the API usage limits of the provider org. *Salesforce Help*: API Usage Considerations for Salesforce Connect—Cross-Org Adapter *Salesforce Limits Quick Reference Guide*: API Requests Limits
OData 2.0 OData 4.0	Uses Open Data Protocol to access data that's stored outside Salesforce. The external data must be exposed via OData producers.	*Salesforce Help*: General Limits for Salesforce Connect—OData 2.0 and 4.0 Adapters
Custom adapter created via Apex	You use the Apex Connector Framework to develop your own custom adapter when the other available adapters aren't suitable for your needs. A custom adapter can obtain data from anywhere. For example, some data can be retrieved from anywhere in the Internet via callouts, while other data can be manipulated or even generated programmatically.	*Apex Developer Guide*: Callout Limits and Limitations *Apex Developer Guide*: Execution Governors and Limits

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

- Salesforce Connect
- OData 2.0 or 4.0 Adapter for Salesforce Connect
- Cross-Org Adapter for Salesforce Connect
- Custom Adapter for Salesforce Connect

Identity Type for External Data Sources

On the external data source, the `Identity Type` field specifies whether your organization uses one set or multiple sets of credentials to access the external system. Each set of credentials corresponds to a login account on the external system.

If you select **Named Principal**, your organization uses only one login account on the external system.

If you select **Per User**, your organization uses multiple login accounts on the external system. Each of your users can have a unique set of credentials, or you can group your users—for example, by function or business unit—and have each group share a set of credentials. After you grant user access to the external data source through permission sets or profiles, users can set up and manage their own authentication settings for the external system.

Tip: Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

	Identity Type is Named Principal	Identity Type is Per User
To access the external system's...	**The system uses the credentials that are defined in the...**	
Data (search or view external objects)	External data source definition	User's personal authentication settings for the external system
Metadata (sync to create external objects)	External data source definition	External data source definition

SEE ALSO:

- Define an External Data Source for Salesforce Connect—Cross-Org Adapter
- Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter
- Define an External Data Source for Salesforce Connect—Custom Adapter
- Grant Access to Authentication Settings for External Data Sources
- Store Authentication Settings for External Systems

Record IDs for Salesforce Connect External Objects

The first time a data row is retrieved from an external system, the external object record is assigned a Salesforce ID. Each record ID remains associated with the same external data row, unless the external object is deleted from the org.

Note: Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.

SEE ALSO:

Record IDs and External IDs for External Objects in Salesforce Connect—Cross-Org Adapter

Salesforce Connect

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Considerations for Salesforce Connect—All Adapters

Salesforce Connect has some special behaviors and limitations.

Also review the considerations for each Salesforce Connect adapter that you use.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

Salesforce Compatibility Considerations for Salesforce Connect—All Adapters

Some Salesforce features and functionality have special behaviors or aren't available for Salesforce Connect external objects.

Sync Considerations for Salesforce Connect—All Adapters

When you validate and sync an external data source, some special behaviors and limitations apply.

Relationship Considerations for Salesforce Connect—All Adapters

Lookup, external lookup, and indirect lookup relationships have some special behaviors and limitations.

Report Considerations for Salesforce Connect—All Adapters

Special behaviors and limitations apply to reports that include external objects.

Record Feed Considerations for Salesforce Connect—All Adapters

Chatter feeds on external objects have some special behaviors and limitations.

API Query Considerations for Salesforce Connect—All Adapters

Special behaviors and limitations apply to `queryAll()` and `queryMore()` calls on external data.

Search Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect searches of external objects.

SOSL Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect SOSL access to external data.

SOQL Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect SOQL access to external data.

Writable External Objects Considerations for Salesforce Connect—All Adapters

Understand the limits and considerations for using Salesforce Connect to create, update, and delete data that's stored outside your Salesforce org.

Packaging Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect packaging of external data sources and their dependent external objects and custom fields.

Lightning Experience Considerations for Salesforce Connect—All Adapters

Users can access external objects from the Lightning Experience app, but some requirements and special behaviors apply.

Salesforce1 Considerations for Salesforce Connect—All Adapters

Users can view and search external objects from the Salesforce1 app, but some special behaviors and limitations apply.

Communities Considerations for Salesforce Connect—All Adapters

Users can access external objects from communities, but some requirements and limitations apply.

General Limits for Salesforce Connect—All Adapters

Understand the limits for using Salesforce Connect.

SEE ALSO:

Considerations for Salesforce Connect—Cross-Org Adapter

Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Considerations for Salesforce Connect—Custom Adapter

Salesforce Compatibility Considerations for Salesforce Connect—All Adapters

Some Salesforce features and functionality have special behaviors or aren't available for Salesforce Connect external objects.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The following features aren't available for external objects.
  - Activities, events, and tasks
  - Apex sharing reasons
  - Apex triggers
  - Approval processes
  - Attachments
  - Field history tracking
  - Flows (Visual Workflow)
  - Merge fields
  - Notes
  - Processes (Process Builder)
  - Record-level security (sharing)
  - Record types
  - Schema Builder
  - Validation rules
  - Workflow rules
- Formulas and roll-up summary fields can't reference fields on external objects.
- The following custom field types aren't available for external objects.
  - Auto-number
  - Currency

  - Formula
  - Geolocation
  - Master-detail relationship
  - Picklist
  - Picklist (multi-select)
  - Roll-up summary
  - Text (encrypted)
  - Text Area (rich)

- The following behavior applies to fields on external objects.
  - Default field values aren't available for custom fields on external objects.
  - External object field values are displayed without leading and trailing spaces. Keep this behavior in mind when you filter by a field that contains leading or trailing spaces.

    For example, include the leading and trailing spaces in the following SOQL query to match the values that are stored in the external system. However, the spaces aren't displayed in the query results.

    ```
    SELECT FirstName__c FROM Buyers__x WHERE FirstName__c = ' Test '
    ```

  - When a user or the Force.com platform edits a custom field on an external object record, leading and trailing spaces are removed from the field value.
  - For text fields on external objects, make sure that the specified length is large enough to contain the external values. If the value from an external system doesn't fit within the length of the associated text field on the external object, the value is truncated in your Salesforce org. To avoid truncated text field values, increase the length for the text field on the external object.
  - For number fields on external objects, ensure that the specified length can contain all digits to the left of the decimal point in external values. If the numeric value from an external system doesn't fit within the length of the associated number field on the external object, the value is blank in your Salesforce org. If you notice blank numeric field values, adjust `Length` and `Decimal Places` for the number field on the external object to accommodate more digits to the left of the decimal point. If the digits to the right of the decimal point in the external values don't fit within the specified decimal places, the value is truncated.
- In change sets, external objects are included in the custom object component.
- In the Metadata API, external objects are represented by the CustomObject metadata type.

Also review the considerations for each Salesforce Connect adapter that you use.

IN THIS SECTION:

Salesforce Compatibility Considerations for Salesforce Connect—Cross-Org Adapter

Some Salesforce features and functionality have special behaviors or aren't available for external objects that are associated with an external data source of type `Salesforce Connect: Cross-Org`.

SEE ALSO:

Salesforce Connect

Considerations for Salesforce Connect—All Adapters

API Usage Considerations for Salesforce Connect—Cross-Org Adapter

Apex Considerations for Salesforce Connect—Custom Adapter

Salesforce Compatibility Considerations for Salesforce Connect—Cross-Org Adapter

Some Salesforce features and functionality have special behaviors or aren't available for external objects that are associated with an external data source of type `Salesforce Connect: Cross-Org`.

- The cross-org adapter for Salesforce Connect can access only queryable objects in the provider org. If you define an external object whose table name specifies an object that can't be queried, your users and the Force.com platform can't access that external object.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Salesforce Compatibility Considerations for Salesforce Connect—All Adapters

Cross-Org Adapter for Salesforce Connect

Sync Considerations for Salesforce Connect—All Adapters

When you validate and sync an external data source, some special behaviors and limitations apply.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Syncing creates or overwrites Salesforce external objects that map to the external system's schema. Syncing doesn't copy any data into your Salesforce org or write data from your org to the external system.
- Each org can have up to 100 external objects. Syncing fails if it causes your org to exceed this limit.
- Object names must be unique across all standard, custom, and external objects in the org. Syncing fails if it tries to create an external object with an API name that conflicts with that of an existing object in the org. If syncing fails for this reason, determine whether the existing object is needed. If not, delete that object, and then sync.

  If the object is needed, change the API name of the existing object to no longer conflict with the table that you're trying to sync. However, if the existing object is an external object that was previously synced, you can't resync it. Manually update the external object and its fields as needed for schema changes on the external system.

  Tip: We recommend that you create your external data sources and external objects in a Developer Edition org. Then use managed packages to deploy the external data sources and external objects to your other orgs. Doing so prevents your external object names from conflicting with other objects in your org by applying a namespace prefix.

- Syncing is a one-time process. If the external system's schema is changed, the changes aren't automatically synced to your Salesforce org. Resync the objects to reflect the changes in the external system.
- Each external object that's created via syncing has its `Deployment Status` set to In Development. When you're ready to expose an external object to users, set its `Deployment Status` to Deployed. Resyncing an external object doesn't change its deployment status.
- When you resync an external object:
  - The `Display URL Reference Field` is set to None.
  - If a custom field has the `Is Name Field` attribute, the attribute is removed. The `External ID` standard field is used as the name field of the external object.
- Syncing doesn't create relationship fields on the external objects in your Salesforce org. However, you can change the field type of a sync-created custom field to Lookup Relationship, External Lookup Relationship, or Indirect Lookup Relationship. Changing the field type of an existing custom field is simpler and more efficient than manually creating a relationship field on the external object.

For example, suppose that the external system has a foreign key relationship. Syncing the related tables creates a text field in your org for the external column that identifies the foreign keys. To reflect the foreign key relationship within your org, change the field type of that text field to External Lookup Relationship.

- A relationship field is a type of custom field. Therefore, like all custom fields on an external object, relationship fields can be overwritten when you sync the external object. See the sync considerations for each Salesforce Connect adapter that you use.
- For text fields on external objects, make sure that the specified length is large enough to contain the external values. If the value from an external system doesn't fit within the length of the associated text field on the external object, the value is truncated in your Salesforce org. To avoid truncated text field values, increase the length for the text field on the external object.
- For number fields on external objects, ensure that the specified length can contain all digits to the left of the decimal point in external values. If the numeric value from an external system doesn't fit within the length of the associated number field on the external object, the value is blank in your Salesforce org. If you notice blank numeric field values, adjust `Length` and `Decimal Places` for the number field on the external object to accommodate more digits to the left of the decimal point. If the digits to the right of the decimal point in the external values don't fit within the specified decimal places, the value is truncated.

Also review the considerations for each Salesforce Connect adapter that you use.

IN THIS SECTION:

Sync Considerations for Salesforce Connect—Cross-Org Adapter

When you validate and sync an external data source of type `Salesforce Connect: Cross-Org`, some special behaviors and limitations apply.

Sync Considerations for Salesforce Connect—OData 2.0 or 4.0 Adapter

When you validate and sync an external data source of type `Salesforce Connect: OData 2.0` or `Salesforce Connect: OData 4.0`, some special behaviors and limitations apply.

Sync Considerations for Salesforce Connect—Custom Adapter

When you validate and sync an external data source of type `Salesforce Connect: Custom`, some special behaviors and limitations apply.

SEE ALSO:

Salesforce Connect

Considerations for Salesforce Connect—All Adapters

Deployment Status for Custom Objects and External Objects

Sync Considerations for Salesforce Connect—Cross-Org Adapter

When you validate and sync an external data source of type `Salesforce Connect: Cross-Org`, some special behaviors and limitations apply.

- Which objects and fields can be synced is determined by the object permissions and field permissions of the provider org's user whose credentials are defined in the external data source definition. See User Access to External Data in Salesforce Connect—Cross-Org Adapter on page 962.
- Only queryable objects can be synced.
- You can't sync an object whose API name contains 38 or more characters. An external object's name can't exceed 40 characters, including the automatically appended `__x` suffix. See API Names for External Objects and Custom Fields in Salesforce Connect—Cross-Org Adapter on page 961.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- These field types aren't synced.
  - Auto number
  - Encrypted text
  - Rich text area
- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.
- Hierarchical, lookup, and master-detail relationship fields are synced. However, in the subscriber org their values appear as IDs, not as record names.

  For example, suppose that we sync the Account object. When we view the Acme Wireless account in the provider org, the value of the Account Owner (`Account.OwnerId`) field appears as John Smith. When we view the equivalent account in the subscriber org, the value for the (`Account__x.OwnerId__c`) field appears as 005B00000019eapIAA.
- The names and labels of synced fields on the subscriber org are derived from the API names—not the labels—of the fields on the provider org. For example, the Account object in the provider org has the Account Owner (`OwnerId`) standard field. If we sync the Account object, the `Account__x.OwnerId__c` field in the subscriber org has the label "Owner ID" and the name "OwnerId."

Important: When you select the provider org objects to sync, determine whether check marks appear in the Synced column.

If a Synced check mark appears, the subscriber org has an external object whose object name (for example, `Account__x`) associates it with the object in the provider org (for example, `Account`). If you select the object and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email_c__c`) associates it with a field on the provider org (for example, `Email__c`).
- Any other custom fields on the external object remain as they are, including:
  - Previously synced custom fields whose API names were changed by editing their `Field Name` values.
  - Manually added custom fields whose API names aren't associated with fields on the provider org's object.

If no Synced check mark appears, and you sync the object, a new external object is created in the subscriber org.

For example, the object name is changed on the provider org to no longer be associated with the object name of the external object on the subscriber org. Syncing that object creates a new external object on the subscriber org. We recommend that you change the object name of the existing external object to match the updated object name on the provider org before you sync.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

API Names for External Objects and Custom Fields in Salesforce Connect—Cross-Org Adapter

Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter

Sync Considerations for Salesforce Connect—All Adapters

Cross-Org Adapter for Salesforce Connect

Sync Considerations for Salesforce Connect—OData 2.0 or 4.0 Adapter

When you validate and sync an external data source of type `Salesforce Connect: OData 2.0` or `Salesforce Connect: OData 4.0`, some special behaviors and limitations apply.

- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.
- For a list of supported OData types, see "OData Type Mapping" in the Salesforce Help.

Important: When you select the tables to sync, determine whether check marks appear in the Synced column.

If a Synced check mark appears, your org has an external object whose object name matches the table name. If you select the table and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email__c`) associates it with a table column name (for example, `Email`).
- Any other custom fields on the external object remain as they are, including:
  - Previously synced custom fields whose API names were changed by editing their `Field Name` values.
  - Manually added custom fields whose API names aren't associated with table column names.

If no Synced check mark appears, and you sync the table, a new external object is created in your org. The new external object's object name matches the table name.

For example, if the table name is changed on the external system to no longer match the object name of the external object, syncing that table creates a new external object in Salesforce. We recommend that you change the object name of the existing external object to match the new table name on the external system before you sync that table.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Sync Considerations for Salesforce Connect—All Adapters

Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

OData 2.0 or 4.0 Adapter for Salesforce Connect

OData Type Mapping

Sync Considerations for Salesforce Connect—Custom Adapter

When you validate and sync an external data source of type `Salesforce Connect: Custom`, some special behaviors and limitations apply.

- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa. For an external data source of type Salesforce Connect: Custom, search is enabled when the custom adapter's `DataSource.Provider` class declares the `DataSource.Capability.SEARCH` capability.

Important: When you select the tables to sync, determine whether check marks appear in the Synced column.

If a Synced check mark appears, your org has an external object whose object name matches the table name. If you select the table and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email__c`) associates it with a table column name (for example, `Email`).
- Any other custom fields on the external object remain as they are, including:
  - Previously synced custom fields whose API names were changed by editing their `Field Name` values.
  - Manually added custom fields whose API names aren't associated with table column names.
- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa. For an external data source of type Salesforce Connect: Custom, search is enabled when the custom adapter's `DataSource.Provider` class declares the `DataSource.Capability.SEARCH` capability.

If no Synced check mark appears, and you sync the table, a new external object is created in your org. The new external object's object name matches the table name.

For example, suppose you change the table name in the custom adapter's `DataSource.Connection` class to no longer match the object name of the external object. Syncing that table creates a new external object in Salesforce. We recommend that you change the object name of the existing external object to match the new table name in the `DataSource.Connection` class before you sync that table.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Sync Considerations for Salesforce Connect—All Adapters

Define an External Data Source for Salesforce Connect—Custom Adapter

Custom Adapter for Salesforce Connect

Relationship Considerations for Salesforce Connect—All Adapters

Lookup, external lookup, and indirect lookup relationships have some special behaviors and limitations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Only lookup, external lookup, and indirect lookup relationships are available for external objects. No other relationship types are supported.
- Depending on the availability of the external system, related lists of child external objects may load slowly when users view the parent record detail pages.
- Relationships that involve external objects allow users to create child records from the record detail pages of parent records. However, the relationship field on each new child record isn't automatically populated to identify the parent record.
- Syncing doesn't create relationship fields on the external objects in your Salesforce org. However, you can change the field type of a sync-created custom field to Lookup Relationship, External Lookup Relationship, or Indirect Lookup Relationship. Changing the field type of an existing custom field is simpler and more efficient than manually creating a relationship field on the external object.

  For example, suppose that the external system has a foreign key relationship. Syncing the related tables creates a text field in your org for the external column that identifies the foreign keys. To reflect the foreign key relationship within your org, change the field type of that text field to External Lookup Relationship.
- A relationship field is a type of custom field. Therefore, like all custom fields on an external object, relationship fields can be overwritten when you sync the external object. See the sync considerations for each Salesforce Connect adapter that you use.
- Cascade-delete isn't available for external object relationships.
- In Salesforce Classic only, external lookup and indirect lookup relationship fields don't display the expected names of parent records.
  - An external lookup field displays either the parent object ID or the value of the parent object's External ID standard field. The latter appears by default. If, however, a custom field on the parent object has the Is Name Field attribute, the parent object ID is displayed.
  - An indirect lookup field displays the value of the target field on the parent object. To find related records, target field values are matched against the values of the indirect lookup relationship field on the child object. The target field, which has the `External ID` and `Unique` attributes, is selected when an indirect lookup field is created.
- When a user tries to edit an external lookup or indirect lookup relationship field, a lookup dialog isn't available for selecting the parent record.
  - To edit an external lookup relationship field, manually enter the value of the `External ID` standard field for the parent record.
  - To edit an indirect lookup relationship field, manually enter the value of the target field of the parent record. The target field is the custom field with `External ID` and `Unique` attributes that was selected when the indirect lookup relationship was created. To determine related records, Salesforce matches target field values against the values of the indirect lookup relationship field on the child object.
- With external lookup and indirect lookup relationships, the parent record appears as a clickable link in the relationship field on the child record. If the child record is viewed by a user who doesn't have access to the parent record, the parent record appears in the relationship field as plain text instead of a link.
- Lookup filters aren't available for external lookup relationship fields.
- Indirect lookup relationship fields can be created on external objects only.
- Only objects that have a custom field with the `External ID` and `Unique` attributes are available as parent objects in indirect lookup relationships. If you don't see the desired object when you create an indirect lookup relationship field, add a custom unique, external ID field to that object.

- If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID**, **Unique**, and **Treat "ABC" and "abc" as different values (case sensitive)**.

SEE ALSO:

External Object Relationships

Sync Considerations for Salesforce Connect—All Adapters

Salesforce Connect

Considerations for Salesforce Connect—All Adapters

Report Considerations for Salesforce Connect—All Adapters

Special behaviors and limitations apply to reports that include external objects.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Depending on network latency and the availability of the external system, reports that include an external object can take a long time to run.
- When you run a report, your org performs a request callout for each external object in the report.
- When you run a report that's in the joined format, your org performs separate request callouts for each block.
- If the URL length of a report callout approaches or exceeds 2 KB, your org splits the request into multiple HTTP calls, with each URL being less than 2 KB. However, with the cross-org adapter for Salesforce Connect, report callouts that exceed 2 KB fail and are not split into multiple calls.
- When a report includes an external object, the report fetches up to 2,000 records for the primary object. This limit applies regardless of whether the primary object is a standard, custom, or external object.
  - If the report has no child objects, the total number of rows is 2,000 or fewer.
  - If the report has a child object, the total number of rows can be greater or less than 2,000, depending on how many child records are fetched.
- As is true for all callouts for external objects, report callouts are limited by the Salesforce Connect adapters in use.

Salesforce Connect Adapter	Description	Where to Find Callout Limits
Cross-org	Uses the Force.com REST API to access data that's stored in other Salesforce orgs.	No callout limits. However, each callout counts toward the API usage limits of the provider org. *Salesforce Help*: API Usage Considerations for Salesforce Connect—Cross-Org Adapter *Salesforce Limits Quick Reference Guide*: API Requests Limits
OData 2.0 OData 4.0	Uses Open Data Protocol to access data that's stored outside Salesforce. The external data must be exposed via OData producers.	*Salesforce Help*: General Limits for Salesforce Connect—OData 2.0 and 4.0 Adapters

Salesforce Connect Adapter	Description	Where to Find Callout Limits
Custom adapter created via Apex	You use the Apex Connector Framework to develop your own custom adapter when the other available adapters aren't suitable for your needs. A custom adapter can obtain data from anywhere. For example, some data can be retrieved from anywhere in the Internet via callouts, while other data can be manipulated or even generated programmatically.	*Apex Developer Guide*: Callout Limits and Limitations *Apex Developer Guide*: Execution Governors and Limits

- For large external data sets, report callouts typically access only a subset of the external data.
  - If the report results in few or no rows, try customizing the report to obtain more relevant external object rows.
  - If the report includes summary fields and formulas, those aggregate values likely reflect only a subset of your data. To improve the accuracy of the aggregate values, try customizing the report to obtain more relevant data.
- The external objects' associated external data sources must have the **High Data Volume** option deselected. This requirement doesn't apply to the cross-org adapter for Salesforce Connect.
- Cross filters don't support external lookup relationships. An external lookup relationship links a child standard, custom, or external object to a parent external object.
- These report features aren't available for external objects.
  - Buckets and bucket fields
  - Historical trend reporting
- Reports can include or reference external objects via lookup, external lookup, and indirect lookup relationship fields. If those relationship fields aren't Salesforce IDs or the name fields of parent objects, the report returns only empty values for those relationship fields.

SEE ALSO:

External Object Relationships

Considerations for Salesforce Connect—All Adapters

Record Feed Considerations for Salesforce Connect—All Adapters

Chatter feeds on external objects have some special behaviors and limitations.

- Field history tracking isn't available for external objects.
- Record feeds aren't available for Salesforce Connect external objects that map to high-data-volume external data sources.

SEE ALSO:

Salesforce Connect

Considerations for Salesforce Connect—All Adapters

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

API Query Considerations for Salesforce Connect—All Adapters

Special behaviors and limitations apply to `queryAll()` and `queryMore()` calls on external data.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

`queryAll()`

Because Salesforce doesn't track changes to external data, `queryAll()` behaves the same as `query()` for external objects.

`queryMore()`

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. When querying external objects, Salesforce Connect accesses the external data in real time via Web service callouts. Each `queryMore()` call results in a Web service callout. The batch boundaries and page sizes depend on your adapter and how you set up the external data source.

We recommend the following:

- When possible, avoid paging by filtering your queries of external objects to return fewer rows than the batch size, which by default is 500 rows. Remember, obtaining each batch requires a `queryMore()` call, which results in a Web service callout.
- If the external data frequently changes, avoid using `queryMore()` calls. If the external data is modified between `queryMore()` calls, you can get an unexpected `QueryResult`.

If the primary or "driving" object for a `SELECT` statement is an external object, `queryMore()` supports only that primary object and doesn't support subqueries.

IN THIS SECTION:

API Query Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Some special behaviors and limitations affect API queries of external data via the OData adapters for Salesforce Connect.

API Query Considerations for Salesforce Connect—Custom Adapter

Some special behaviors and limitations affect API queries of external data via custom adapters for Salesforce Connect.

SEE ALSO:

SOQL Considerations for Salesforce Connect—All Adapters

Salesforce Connect

Considerations for Salesforce Connect—All Adapters

API Query Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Some special behaviors and limitations affect API queries of external data via the OData adapters for Salesforce Connect.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

By default, the OData 2.0 and 4.0 adapters for Salesforce Connect use client-driven paging. With client-driven paging, OData adapters convert each `queryMore()` call into an OData query that uses the `$skip` and `$top` system query options to specify the batch boundary and page size. These options are similar to using `LIMIT` and `OFFSET` clauses to page through a result set.

If you enable server-driven paging on an external data source, Salesforce ignores the requested page sizes, including the default `queryMore()` batch size of 500 rows. The pages returned by the external system determine the batches.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

- API Query Considerations for Salesforce Connect—All Adapters
- OData Query String Options
- OData 2.0 or 4.0 Adapter for Salesforce Connect
- Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

API Query Considerations for Salesforce Connect—Custom Adapter

Some special behaviors and limitations affect API queries of external data via custom adapters for Salesforce Connect.

Custom adapters for Salesforce Connect, which you create with the Apex Connector Framework, don't automatically support `queryMore()` or paging of any kind. You can implement `queryMore()` and client-driven paging by using `LIMIT` and `OFFSET` clauses to page through result sets.

If you enable server-driven paging on an external data source, Salesforce ignores the requested page sizes, including the default `queryMore()` batch size of 500 rows. The pages returned by the external system determine the batches. To enable server-driven paging for a custom adapter, the `DataSource.Provider` class must declare the `QUERY_PAGINATION_SERVER_DRIVEN` capability on the `getCapabilities` method. Also, the Apex code must generate a query token and use it to determine and fetch the next batch of results.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

- API Query Considerations for Salesforce Connect—All Adapters
- Custom Adapter for Salesforce Connect
- *Apex Developer Guide*

Search Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect searches of external objects.

- To include an external object in SOSL and Salesforce searches, enable search on both the external object and the external data source. However, syncing always overwrites the external object's search status to match the search status of the external data source.
- (Salesforce Classic only) External object search results display only the top 25 rows.
- Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.
- We recommend that you set up the external system to interpret the search query strings as you prefer. For example, perhaps your users want to search for case-insensitive strings.

Also review the considerations for each Salesforce Connect adapter that you use.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

SEE ALSO:

Search Considerations for Salesforce Connect—OData 2.0 Adapter

Some special behaviors and limitations affect searches of external data via the OData 2.0 adapter for Salesforce Connect.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

We send the entire search query string to the external system as a case-sensitive single phrase after removing all ASCII punctuation characters except hyphens (-). For example, if a user searches for `Sales & Marketing`, the external system receives "Sales Marketing" as a case-sensitive search query string.

Also, by default, the search query string is used as the `substringof` value in the `$filter` system query option. In the following example, the search query string is `Acme`.

```
http://services.example.org/my.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$inlinecount=allpages&
$filter=substringof('Acme',CompanyName)+eq+true+
or+substringof('Acme',Phone)+eq+true&$top=26
```

We recommend that you implement a free-text-search custom query option on the OData producer. Specify the name of that query string parameter in the `Custom Query Option for Salesforce Search` field on the external data source. In the following example, the custom query parameter is named `doSearch` and the search query string is `Acme`.

```
http://services.example.org/my.svc/Shippers?
$select=CompanyName,Phone,ShipperID&
$inlinecount=allpages&doSearch=Acme&$top=26
```

Learn more about OData URI conventions at `www.odata.org`.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

Search Considerations for Salesforce Connect—OData 4.0 Adapter

Some special behaviors and limitations affect searches of external data via the OData 4.0 adapter for Salesforce Connect.

We send the entire search query string to the external system as a case-sensitive single phrase after removing all ASCII punctuation characters except hyphens (-). For example, if a user searches for *`Sales & Marketing`*, the external system receives "Sales Marketing" as a case-sensitive search query string.

Also, by default, the search query string is used as the `contains` value in the `$filter` system query option. In the following example, the search query string is *`Acme`*.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

```
http://services.example.org/v4.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$count=true&
$filter=contains(CompanyName,'Acme') eq true
or contains(Phone,'Acme') eq true$top=26&
```

We recommend that you set up the OData producer to support free-text search expressions with the `$search` system query option. Then select `Use Free-Text Search Expressions` on the external data source. In the following example, the search query string is *`Acme`*.

```
http://services.example.org/v4.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$count=true&
$search=Acme&$top=26
```

Learn more about OData URI conventions at `www.odata.org`.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

- Search Considerations for Salesforce Connect—All Adapters
- OData Query String Options
- OData 2.0 or 4.0 Adapter for Salesforce Connect

SOSL Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect SOSL access to external data.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- To include an external object in SOSL and Salesforce searches, enable search on both the external object and the external data source. However, syncing always overwrites the external object's search status to match the search status of the external data source.
- Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.
- External objects don't support the following.
  - `INCLUDES` operator
  - `LIKE` operator
  - `EXCLUDES` operator

  - `toLabel()` function
- External objects also don't support Salesforce Knowledge-specific clauses, including the following.
  - `UPDATE TRACKING` clause
  - `UPDATE VIEWSTAT` clause
  - `WITH DATA CATEGORY` clause
- External objects must be specified explicitly in a `RETURNING` clause to be returned in search results. For example:

```
FIND {MyProspect} RETURNING MyExternalObject, MyOtherExternalObject
```

Also review the considerations for each Salesforce Connect adapter that you use.

IN THIS SECTION:

SOSL Considerations for Salesforce Connect—OData 2.0 Adapter
Some special behaviors and limitations affect SOSL access to external data via the OData 2.0 adapter for Salesforce Connect.

SOSL Considerations for Salesforce Connect—OData 4.0 Adapter
Some special behaviors and limitations affect SOSL access to external data via the OData 4.0 adapter for Salesforce Connect.

SOSL Considerations for Salesforce Connect—Custom Adapter
Some special behaviors and limitations affect SOSL access to external data via custom adapters for Salesforce Connect.

SEE ALSO:

Considerations for Salesforce Connect—All Adapters
Force.com SOQL and SOSL Reference
Sync Considerations for Salesforce Connect—All Adapters
Salesforce Connect

SOSL Considerations for Salesforce Connect—OData 2.0 Adapter

Some special behaviors and limitations affect SOSL access to external data via the OData 2.0 adapter for Salesforce Connect.

The OData adapters for Salesforce Connect don't support logical operators in a `FIND` clause. We send the entire search query string to the external system as a case-sensitive single phrase after removing all ASCII punctuation characters except hyphens (-). For example, `FIND {MyProspect OR "John Smith"}` searches for the exact phrase "MyProspect OR John Smith".

Also, by default, the search query string is used as the `substringof` value in the `$filter` system query option. In the following example, the search query string is *`Acme`*.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

```
http://services.example.org/my.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$inlinecount=allpages&
$filter=substringof('Acme',CompanyName)+eq+true+
or+substringof('Acme',Phone)+eq+true&$top=26
```

We recommend that you implement a free-text-search custom query option on the OData producer. Specify the name of that query string parameter in the `Custom Query Option for Salesforce Search` field on the external data source. In the following example, the custom query parameter is named `doSearch` and the search query string is *Acme*.

```
http://services.example.org/my.svc/Shippers?
$select=CompanyName,Phone,ShipperID&
$inlinecount=allpages&doSearch=Acme&$top=26
```

Learn more about OData URI conventions at www.odata.org.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

- OData 2.0 or 4.0 Adapter for Salesforce Connect
- SOSL Considerations for Salesforce Connect—All Adapters
- *Force.com SOQL and SOSL Reference*
- OData Query String Options
- Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

SOSL Considerations for Salesforce Connect—OData 4.0 Adapter

Some special behaviors and limitations affect SOSL access to external data via the OData 4.0 adapter for Salesforce Connect.

The OData adapters for Salesforce Connect don't support logical operators in a `FIND` clause. We send the entire search query string to the external system as a case-sensitive single phrase after removing all ASCII punctuation characters except hyphens (-). For example, `FIND {MyProspect OR "John Smith"}` searches for the exact phrase "MyProspect OR John Smith".

Also, by default, the search query string is used as the `contains` value in the `$filter` system query option. In the following example, the search query string is *Acme*.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

```
http://services.example.org/v4.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$count=true&
$filter=contains(CompanyName,'Acme') eq true
or contains(Phone,'Acme') eq true$top=26&
```

We recommend that you set up the OData producer to support free-text search expressions with the `$search` system query option. Then select `Use Free-Text Search Expressions` on the external data source. In the following example, the search query string is *Acme*.

```
http://services.example.org/v4.svc/Shippers?
$select=CompanyName,Phone,ShipperID&$count=true&
$search=Acme&$top=26
```

Learn more about OData URI conventions at www.odata.org.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

- OData 2.0 or 4.0 Adapter for Salesforce Connect
- SOSL Considerations for Salesforce Connect—All Adapters
- *Force.com SOQL and SOSL Reference*
- OData Query String Options
- Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

SOSL Considerations for Salesforce Connect—Custom Adapter

Some special behaviors and limitations affect SOSL access to external data via custom adapters for Salesforce Connect.

- The `convertCurrency()` function isn't supported in SOSL queries of external objects.
- `WITH` clauses aren't supported in SOSL queries of external objects.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

- SOSL Considerations for Salesforce Connect—All Adapters
- *Force.com SOQL and SOSL Reference*
- Custom Adapter for Salesforce Connect

SOQL Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect SOQL access to external data.

- A subquery that involves external objects can fetch up to 1,000 rows of data.
- Each SOQL query can have up to 4 joins across external objects and other types of objects.

  Each join requires a separate round trip to the external system when executing the query. Expect longer response times for each join in a query.
- External objects don't support the following aggregate functions and clauses.
  - `AVG()` function
  - `COUNT(`***`fieldName`***`)` function (however, `COUNT()` is supported)
  - `HAVING` clause
  - `GROUP BY` clause
  - `MAX()` function
  - `MIN()` function
  - `SUM()` function
- External objects also don't support the following.
  - `EXCLUDES` operator
  - `FOR VIEW` clause
  - `FOR REFERENCE` clause

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- `INCLUDES` operator
- `LIKE` operator
- `toLabel()` function
- `TYPEOF` clause
- `WITH` clause

Also review the considerations for each Salesforce Connect adapter that you use.

IN THIS SECTION:

SOQL Considerations for Salesforce Connect—OData 2.0 Adapter
Some special behaviors and limitations affect SOQL access to external data via the OData 2.0 adapter for Salesforce Connect.

SOQL Considerations for Salesforce Connect—Custom Adapter
Some special behaviors and limitations affect SOQL access to external data via custom adapters for Salesforce Connect.

SEE ALSO:

SOSL Considerations for Salesforce Connect—All Adapters

SOQL Considerations for Salesforce Connect—OData 2.0 Adapter

Some special behaviors and limitations affect SOQL access to external data via the OData 2.0 adapter for Salesforce Connect.

- External objects have the following limitations for the `ORDER BY` clause.
  - `NULLS FIRST` and `NULLS LAST` are ignored.
  - External objects don't support the `ORDER BY` clause in relationship queries.
- The `COUNT()` aggregate function is supported only on external objects whose external data sources have `Request Row Counts` enabled. Specifically, the response from the external system must include the total row count of the result set.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

SOQL Considerations for Salesforce Connect—All Adapters
Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter
OData 2.0 or 4.0 Adapter for Salesforce Connect

SOQL Considerations for Salesforce Connect—Custom Adapter

Some special behaviors and limitations affect SOQL access to external data via custom adapters for Salesforce Connect.

- Location-based SOQL queries of external objects aren't supported.
- If a SOQL query of an external object includes the following, the query fails.
  - `convertCurrency()` function
  - `UPDATE TRACKING` clause
  - `UPDATE VIEWSTAT` clause
  - `USING SCOPE` clause
- In an `ORDER BY` clause, the following are ignored.
  - `NULLS FIRST` syntax
  - `NULLS LAST` syntax

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

SOQL Considerations for Salesforce Connect—All Adapters

Custom Adapter for Salesforce Connect

Writable External Objects Considerations for Salesforce Connect—All Adapters

Understand the limits and considerations for using Salesforce Connect to create, update, and delete data that's stored outside your Salesforce org.

- The external system handles write conflicts, so we can't guarantee that all write operations that are initiated from within Salesforce are applied.
- Write operations that are initiated from different contexts can occur in varying order.
- The API can't save changes to an external object and a standard or custom object within the same transaction.
- Depending on how the external system executes write operations from Salesforce, it can take some time for changes to external object records to take effect. If you don't see recent changes when you view or query an external object record, try again later.
- When a user or the Force.com platform edits a custom field on an external object record, leading and trailing spaces are removed from the field value.
- When a user tries to edit an external lookup or indirect lookup relationship field, a lookup dialog isn't available for selecting the parent record.
  - To edit an external lookup relationship field, manually enter the value of the `External ID` standard field for the parent record.
  - To edit an indirect lookup relationship field, manually enter the value of the target field of the parent record. The target field is the custom field with `External ID` and `Unique` attributes that was selected when the indirect lookup relationship was created. To determine related records, Salesforce matches target field values against the values of the indirect lookup relationship field on the child object.

Also review the considerations for each Salesforce Connect adapter that you use.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

SEE ALSO:

Salesforce Connect
Considerations for Salesforce Connect—All Adapters

Writable External Objects Aren't Available for Salesforce Connect—Cross-Org Adapter

Writable external objects aren't supported with the cross-org adapter for Salesforce Connect.

Your users can't create, edit, or delete external object records through the cross-org adapter for Salesforce Connect. However, you can enable your users to log in to the provider org and update the records there.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Writable External Objects Considerations for Salesforce Connect—All Adapters
Cross-Org Adapter for Salesforce Connect

Writable External Objects Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Some special behaviors and limitations affect writable external objects that are associated with OData adapters for Salesforce Connect.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- An external object custom field associated with an OData complex type on the external system is always read only, even if the external object is writable.
- Writable external objects aren't available for high-data-volume external data sources.
- Make sure that the OData producer supports these HTTP verbs: `POST`, `PUT`, `MERGE`, `PATCH`, and `DELETE`.
- When a user edits fields on an external object record from the Salesforce user interface, Salesforce Connect sends an HTTP `POST` request to the external system. That `POST` request includes all the external object's fields, including the fields that the user didn't change.
- If an external object record is edited from the API, only the specified fields are included in the HTTP `POST` request.
- If your external system requires write operations to specify values for entity keys, ensure that `External ID` standard field values and entity key values don't contradict each other. For each write operation, include either the External ID standard field value or the custom field values that form the entity key, but never both.

Also review the considerations that apply to all Salesforce Connect adapters.

SEE ALSO:

Writable External Objects Considerations for Salesforce Connect—All Adapters

External IDs and OData Entity Keys

OData 2.0 or 4.0 Adapter for Salesforce Connect

Writable External Objects Considerations for Salesforce Connect—Custom Adapters

Some special behaviors and limitations affect writable external objects that are associated with custom adapters for Salesforce Connect.

- Writable external objects aren't available for high-data-volume external data sources.
- Queued changes to the external data execute over time, so external object records that are read successively can contain different data.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Writable External Objects Considerations for Salesforce Connect—All Adapters

Custom Adapter for Salesforce Connect

Apex Developer Guide: Writable External Objects

Packaging Considerations for Salesforce Connect—All Adapters

Some special behaviors and limitations affect packaging of external data sources and their dependent external objects and custom fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- After installing an external data source from a managed or unmanaged package, the subscriber must re-authenticate to the external system.
  - For password authentication, the subscriber must re-enter the password in the external data source definition.
  - For OAuth, the subscriber must update the callback URL in the client configuration for the authentication provider, then re-authenticate by selecting `Start Authentication Flow on Save` on the external data source.
- Certificates aren't packageable. If you package an external data source that specifies a certificate, make sure that the subscriber org has a valid certificate with the same name.
- In managed and unmanaged packages, external objects are included in the custom object component.
- If you add an external data source to a package, no other components are automatically included in the package.
- If you add an external object to a package, then list views, page layouts, custom fields, the custom object component that defines the external object, and the external data source are automatically included in the package. If the external object has lookup, external lookup, or indirect lookup relationship fields, the parent objects are also automatically included in the package.
- If external data source access is assigned in a permission set or in profile settings that you add to a package, the enabled external data sources are automatically included in the package.

- If an external object is assigned in a permission set or in profile settings that you add to a package, then custom fields, list views, page layouts, the custom object component that defines the external object, and the external data source are automatically included in the package.

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Salesforce Connect

Lightning Experience Considerations for Salesforce Connect—All Adapters

Users can access external objects from the Lightning Experience app, but some requirements and special behaviors apply.

- The external objects' associated external data sources must have the **High Data Volume** option deselected. This requirement doesn't apply to the cross-org adapter for Salesforce Connect.
- Quick actions on external objects aren't supported.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Salesforce Connect

Salesforce1 Considerations for Salesforce Connect—All Adapters

Users can view and search external objects from the Salesforce1 app, but some special behaviors and limitations apply.

- As long as the following conditions are true, external objects are included in the Recent section of the Salesforce1 navigation menu as part of the Smart Search Items element. Users might need to tap **More** to see these items.
  - As with custom objects, external objects must be assigned to tabs that users can access, and object permissions must be granted via profiles or permission sets.
  - The external objects' associated external data sources must have the **High Data Volume** option deselected. This requirement doesn't apply to the cross-org adapter for Salesforce Connect.
- To allow external objects to appear in the following apps, select **Enable Search** in the associated external data sources. This requirement doesn't apply to custom adapters for Salesforce Connect.
  - Salesforce1 downloadable app for iOS
  - Salesforce1 mobile browser app when used on iOS devices

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Salesforce Connect

Communities Considerations for Salesforce Connect—All Adapters

Users can access external objects from communities, but some requirements and limitations apply.

- You must build the communities with the Customer Service (Napili) template or the Salesforce Tabs + Visualforce template.
- This list applies to communities that are built with the Customer Service (Napili) template.
  - The external objects' associated external data sources must have the **High Data Volume** option deselected. This requirement doesn't apply to the cross-org adapter for Salesforce Connect.
  - If an external data source's `Identity Type` is set to Per User, community members can't set up their own credentials. However, you can set up and manage each user's authentication settings for external systems from Lightning Experience or Salesforce Classic.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Salesforce Connect

General Limits for Salesforce Connect—All Adapters

Understand the limits for using Salesforce Connect.

For each user, the maximum number of external objects to which you can grant object permissions is equal to the maximum number of custom objects that the user is allowed to access. This limit is determined by the user license that's assigned to the user.

Maximum external objects per org[1]	100
Maximum joins per query across external objects and other types of objects	4
Maximum length of the OAuth token that's issued by the external system	4,000 characters
Maximum new rows retrieved or created per hour per external data source. This limit doesn't apply to high-data-volume external data sources or to rows that have previously been retrieved.	50,000

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[1] The limit of 100 external objects applies regardless of how many Salesforce Connect add-ons you purchase for your org.

IN THIS SECTION:

General Limits for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the limits for the OData adapters for Salesforce Connect.

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Salesforce Connect

General Limits for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the limits for the OData adapters for Salesforce Connect.

An org is limited to:

- 10,000 OData callouts per hour for Enterprise, Performance, and Unlimited Editions. Higher limits are available on request.
- 1,000 OData callouts per hour for Developer Edition.

Maximum HTTP request size for OData	8 MB
Maximum HTTP response size for OData	8 MB
Maximum result set size for an OData query	16 MB
Maximum result set size for an OData subquery	1,000 rows

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

General Limits for Salesforce Connect—All Adapters

OData Callout Rate Limit Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

OData 2.0 or 4.0 Adapter for Salesforce Connect

Salesforce Connect: Cross-Org—Access Data in Another Salesforce Organization

Cross-Org Adapter for Salesforce Connect

Collaborate more effectively and improve processes by connecting the data across your Salesforce orgs. With the cross-org adapter, Salesforce Connect uses Force.com REST API calls to access records in other Salesforce orgs. Nevertheless, setup is quick and easy with point-and-click tools.

Your users and the Force.com platform interact with other orgs' data via external objects. The cross-org adapter for Salesforce Connect converts each of those interactions into a Force.com REST API call.

Suppose that you store your inventory of products in one Salesforce org. You want your regional and local branch offices, who have their own orgs, to see the latest information about your stock. With the cross-org adapter for Salesforce Connect, those other organizations can easily access your data while respecting access restrictions that you control.

The cross-org adapter makes a Force.com REST API call each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record. (This doesn't apply to the cross-org adapter.)
- A user runs a report.
- The preview loads in the report builder.
- An external object is queried via flows, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

To set up Salesforce Connect with the cross-org adapter, you use only point-and-click tools.

IN THIS SECTION:

Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter

The *provider org* stores the data that the *subscriber org* accesses.

API Names for External Objects and Custom Fields in Salesforce Connect—Cross-Org Adapter

If external objects and custom fields are created in the subscriber org via syncing, their API names are derived from the corresponding API names in the provider org.

Record IDs and External IDs for External Objects in Salesforce Connect—Cross-Org Adapter

External object record IDs are derived from the corresponding record IDs in the provider organization. External ID values in external object records match the record IDs in the provider organization.

User Access to External Data in Salesforce Connect—Cross-Org Adapter

A user's access to external data is determined by settings on both subscriber and provider orgs.

SEE ALSO:

Salesforce Connect

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Considerations for Salesforce Connect—Cross-Org Adapter

Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter

The *provider org* stores the data that the *subscriber org* accesses.

You define the external data source and external objects in the subscriber org. Manually create the external objects and their fields, or automatically create them by syncing the provider org's metadata. When users view or search those external objects in the subscriber org, the data is obtained from the provider org and displayed in the subscriber org.

- An org can serve as both a subscriber and a provider.
- A subscriber org can access data from multiple provider orgs.
- A provider org can let multiple subscriber orgs access its data.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Cross-Org Adapter for Salesforce Connect

API Names for External Objects and Custom Fields in Salesforce Connect—Cross-Org Adapter

If external objects and custom fields are created in the subscriber org via syncing, their API names are derived from the corresponding API names in the provider org.

Each external object's API name ends with `__x`. Custom fields on external objects use the traditional `__c` suffix in the API name. Specifically for objects and custom fields that are synced with the cross-org adapter for Salesforce Connect:

- For an API name with no suffix in the provider org, the API name is reused in the subscriber org, but with an applied `__x` suffix for an object or `__c` suffix for a field.
- For an API name with a suffix in the provider org, the API name is reused in the subscriber org. But one of the underscores (_) from the original suffix is removed, and a new `__x` or `__c` suffix is applied.

Example: If you sync the provider org's Account object, the subscriber org creates:

- An external object with the API name `Account__x`
- Custom fields including one with the API name `Account__x.Name__c`

If you sync the provider org's `CustObj__c` object, the subscriber org creates:

- An external object with the API name `CustObj_c__x`
- Custom fields including one with the API name `CustObj_c__x.Name__c`

If the provider org's object has a custom field, the subscriber org creates the custom field on the equivalent external object, for example:

- `Account__x.MyCustField_c__c`
- `CustObj_c__x.MyOtherCustField_c__c`

If you sync the provider org's `Account__x` external object, the subscriber org creates:

- An external object with the API name `Account_x__x`
- Custom fields including one with API name `Account_x__x.Name_c__c`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Cross-Org Adapter for Salesforce Connect

Record IDs and External IDs for External Objects in Salesforce Connect—Cross-Org Adapter

External object record IDs are derived from the corresponding record IDs in the provider organization. External ID values in external object records match the record IDs in the provider organization.

Each object in Salesforce has an object ID with a key prefix as the first three characters. When an external object is created, it's assigned a unique key prefix.

Each external object record has a record ID that uses the same key prefix as the external object ID. The rest of the external object record ID matches the original record ID that's in the provider organization, excluding its original key prefix.

Each record ID that comes from the provider organization becomes a case-insensitive 18-character alphanumeric string in the subscriber organization.

The original record ID is available in the subscriber organization as the value of the External ID standard field on the external object record.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each external object has an `External ID` standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

Example: You sync the provider organization's Account object, and the subscriber organization's `Account__x` object is assigned the key prefix `x00`. An account in the provider organization with the ID 001B0000003SVC7IAO appears in the subscriber organization with the ID x00B0000003SVC7IAO and the external ID 001B0000003SVC7IAO.

SEE ALSO:

Cross-Org Adapter for Salesforce Connect

User Access to External Data in Salesforce Connect—Cross-Org Adapter

A user's access to external data is determined by settings on both subscriber and provider orgs.

The credentials that are used by the subscriber org to connect to the provider org are associated with a user in the provider org. We refer to this user as the *connected user*.

A user in the subscriber org can access only data that the connected user can access within the provider org. In other words, the subscriber org's user access respects the connected user's access restrictions, which are determined by these settings in the provider org.

- Object-level security—permission sets and profiles
- Field-level security—permission sets and profiles
- Record-level security—organization-wide sharing settings, role hierarchies, and sharing rules

In the subscriber org, grant users access to external objects via permission sets and profiles.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Cross-Org Adapter for Salesforce Connect

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

USER PERMISSIONS

To create and edit external data sources:	"Customize Application"
To create and edit external objects:	"Customize Application"
To define or change object-level help:	"Customize Application"
To create and edit custom fields:	"Customize Application"
To edit permission sets and user profiles:	"Manage Profiles and Permission Sets"
To edit another user's authentication settings for external systems:	"Manage Users"

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Provide users with a seamless view of data in your other Salesforce orgs so that they have a complete view of the business. Setting up the cross-org adapter for Salesforce Connect is quick and easy with point-and-click tools.

Setting up Salesforce Connect with the cross-org adapter involves these high-level steps.

1. Define an external data source of type `Salesforce Connect: Cross-Org`.

   Create an external data source for each provider org.

2. Create the external objects.

   Perform this task only if you don't sync to automatically create the external objects. In the subscriber org, create an external object for each object in the provider org that you want to access.

3. Create help content for the external objects.

   Help your users distinguish between external objects and the other objects in the subscriber org, which can have similar names and types of data. On the subscriber org, create Visualforce pages to describe the external objects. When your users click **Help for this Page** on an external object, they read your custom help content.

4. Add custom fields and relationships to the external objects.

   Create relationships between objects. If you didn't sync to automatically create the external objects and their fields on the subscriber org, create a custom field for each of the provider org's fields that you want to access.

5. Enable user access to external objects.

   Grant object permissions through permission sets or profiles.

6. Enable user access to the fields on the external objects.

   Grant field permissions through permission sets or profiles.

7. If the external data source uses per-user authentication:

   a. Let users authenticate to the external system.

      Grant users access to authentication settings for the external data source through permission sets or profiles.

   b. Set up each user's authentication settings.

      You or your users can perform this task.

      Tip: Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for the provider org. If you're using OAuth 2.0, the OAuth flow displays the Salesforce login page twice: first to log in to the provider org to obtain an access token, and then to log back in to the subscriber org. Test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

SEE ALSO:

Cross-Org Adapter for Salesforce Connect

Considerations for Salesforce Connect—Cross-Org Adapter

Developer Guide: *Visualforce Developer Guide*

External Object Relationships

Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter

Define an External Data Source for Salesforce Connect—Cross-Org Adapter

Give your users a seamless view of data across your Salesforce orgs.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. If you set Identity Type to Per User, this label appears when your users view or edit their authentication settings for external systems.
Name	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Type	Select **Salesforce Connect: Cross-Org**.
Connect to	Determines which URL is used to connect to the provider org.
URL	If you selected Connect to Custom URL, enter the login URL for the provider org.
API Version	Select an API version that the provider org supports. The API version determines which of the provider org's objects, fields, and types you can access from the subscriber org.
Connection Timeout	Number of seconds to wait for a response from the provider org before timing out. By default, the value is set to the maximum of 120 seconds.
Enable Search	Determines whether global searches in the subscriber org also search the external objects' data, which is stored in the provider org. When selected, you can control which external objects are searchable by selecting or deselecting Allow Search on each external object. Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.
Identity Type	Determines whether the subscriber org uses one set or multiple sets of credentials to access the provider org. See Identity Type for External Data Sources on page 934.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit external data sources:

- "Customize Application"

4. Select the authentication protocol.
   - If you select **Password Authentication**, enter the username and password for accessing the external system.
   - If you select **OAuth 2.0**, complete the following fields.

Field	Description
`Authentication Provider`	Select a Salesforce authentication provider. See "Configure a Salesforce Authentication Provider" in the Salesforce Help.
`Scope`	Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Using the Scope Parameter. Note: – The value that you enter replaces the `Default Scopes` value that's defined in the specified authentication provider. – Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. – We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.
`Start Authentication Flow on Save`	To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow. When you click **Save**, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org. Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the `Scope` or `Authentication Provider` fields.

5. Click **Save**.
6. Click **Validate and Sync**, and confirm that the connection is successful.
   If you instead receive an error message, refer to the following documents.
   - "Status Codes and Error Responses" in the *Force.com REST API Developer Guide*
   - The "API Fault Element," "ExceptionCode," "Error," and "StatusCode" sections of "Core Data Types Used in API Calls" in the *SOAP API Developer Guide*
7. Optionally, select tables and click **Sync** to do the following for each selected table.
   - Automatically create a Salesforce external object.
   - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.

   Note: Before you sync, make sure that you understand the considerations that are described in these topics.
   - Sync Considerations for Salesforce Connect—All Adapters on page 938
   - Sync Considerations for Salesforce Connect—Cross-Org Adapter on page 939

You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

SEE ALSO:

- Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect
- Store Authentication Settings for External Systems
- API Names for External Objects and Custom Fields in Salesforce Connect—Cross-Org Adapter
- Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter
- Developer Guide: *Force.com REST API Developer Guide*

Considerations for Salesforce Connect—Cross-Org Adapter

Understand the special behaviors, limits, and recommendations for using the cross-org adapter for Salesforce Connect.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

API Usage Considerations for Salesforce Connect—Cross-Org Adapter
With the cross-org adapter, Salesforce Connect uses Force.com REST API calls to access records in other Salesforce orgs. Depending on how the external object is accessed, each call counts toward the API usage limits of only the provider org or of both provider and subscriber orgs.

SEE ALSO:

- Considerations for Salesforce Connect—All Adapters

API Usage Considerations for Salesforce Connect—Cross-Org Adapter

With the cross-org adapter, Salesforce Connect uses Force.com REST API calls to access records in other Salesforce orgs. Depending on how the external object is accessed, each call counts toward the API usage limits of only the provider org or of both provider and subscriber orgs.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

When a user accesses an external object in one of the following ways, the Force.com REST API call counts toward the API usage limits of the provider org.

- Opening a list view of external object records
- Viewing an external object record detail page
- Viewing a parent object record that contains an external object related list
- Executing a search that also searches external objects
- Accessing an external object from a flow, Visualforce page, Apex class, or Apex trigger
- Editing an external object record's details
- Running a report
- Editing a report and causing the preview to load in report builder

If a user or system accesses an external object through the SOAP API, Bulk API, or Force.com REST API, that access counts toward the API usage limits of both the subscriber org and the provider org.

SEE ALSO:

Salesforce Limits Quick Reference Guide: API Requests Limits

Considerations for Salesforce Connect—Cross-Org Adapter

Subscriber and Provider Orgs in Salesforce Connect—Cross-Org Adapter

Salesforce Connect: OData 2.0—Access External Data via the Open Data Protocol

OData 2.0 or 4.0 Adapter for Salesforce Connect

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Connect to your back office for a complete view of your business. With the OData 2.0 or 4.0 adapter, Salesforce Connect uses Open Data Protocol Version 2.0 or Version 4.0 to access data that's stored outside Salesforce.

Your users and the Force.com platform interact with the external data via external objects. Salesforce Connect converts each of those interactions into an OData query that contains the relevant parameters to filter the results. Salesforce performs an OData callout each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record. (This doesn't apply to the cross-org adapter.)
- A user runs a report.
- The preview loads in the report builder.
- An external object is queried via flows, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

The OData 2.0 adapter for Salesforce Connect can access external data that's exposed via services called *OData producers*. Learn more about OData producers at `www.odata.org`.

IN THIS SECTION:

External IDs and OData Entity Keys

When you access external data with the OData 2.0 or 4.0 adapter for Salesforce Connect, the values of the External ID standard field on an external object are derived according to the entity key that's defined in the OData service metadata document.

Client-driven and Server-driven Paging for Salesforce Connect—OData 2.0 and 4.0 Adapters

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You decide whether to have the paging behavior controlled by the external system (server-driven) or by the OData 2.0 or 4.0 adapter for Salesforce Connect (client-driven).

SEE ALSO:

Salesforce Connect

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

OData Reference for Salesforce Connect—OData 2.0 and 4.0 Adapters

External IDs and OData Entity Keys

When you access external data with the OData 2.0 or 4.0 adapter for Salesforce Connect, the values of the External ID standard field on an external object are derived according to the entity key that's defined in the OData service metadata document.

Each external object has an `External ID` standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

Important: Don't use sensitive data as the values of the External ID standard field, because Salesforce sometimes stores those values.

- External lookup relationship fields on child records store and display the External ID values of the parent records.
- For internal use only, Salesforce stores the External ID value of each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.

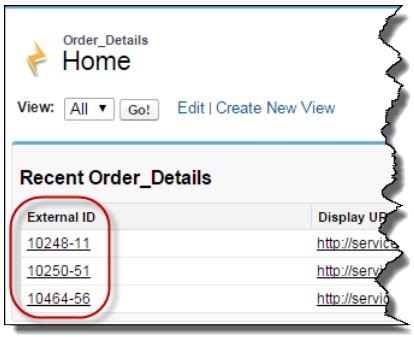
EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

This list view for the Order_Detail external object displays External ID values.



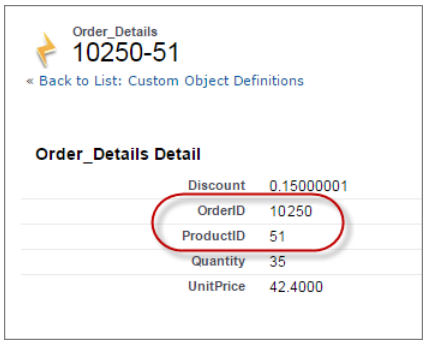
Each External ID value is derived according to the entity key that's defined in the OData service metadata document of the remote data service (OData producer). The entity key is formed from a subset of the entity type's properties.

This excerpt from an OData service metadata document shows that the External ID values for the Order_Detail external object are derived from the OrderID and ProductID properties.

```
<EntityType Name="Order_Detail">
  <Key>
    <PropertyRef Name="OrderID"/>
    <PropertyRef Name="ProductID"/>
  </Key>
  <Property Name="OrderID" Type="Edm.Int32" Nullable="false"/>
  <Property Name="ProductID" Type="Edm.Int32" Nullable="false"/>
  <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="false" Precision="19" Scale="4"/>

  <Property Name="Quantity" Type="Edm.Int16" Nullable="false"/>
  <Property Name="Discount" Type="Edm.Single" Nullable="false"/>
...
```

This record detail page displays the OrderID and ProductID fields. Their values are combined to create the value of the External ID standard field.



If you enable writable external objects, determine whether the external system requires write operations to specify values for the entity keys. For example, many external systems generate values for entity keys when new external object records are created in Salesforce. If your external system requires write operations to specify values for entity keys, ensure that `External ID` standard field values and entity key values don't contradict each other. For each write operation, include either the External ID standard field value or the custom field values that form the entity key, but never both.

SEE ALSO:

- OData 2.0 or 4.0 Adapter for Salesforce Connect
- Writable External Objects Considerations for Salesforce Connect—All Adapters

Client-driven and Server-driven Paging for Salesforce Connect—OData 2.0 and 4.0 Adapters

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You decide whether to have the paging behavior controlled by the external system (server-driven) or by the OData 2.0 or 4.0 adapter for Salesforce Connect (client-driven).

By default, the OData 2.0 and 4.0 adapters for Salesforce Connect use client-driven paging. Specifically, the OData requests use the `$top` and `$skip` system query options to page through the result set.

With server-driven paging, the external system determines the page sizes and batch boundaries. The external system's paging settings can optimize the external system's performance and improve the load times for external objects in your org. Also, the external data set can change while your users or the Force.com platform are paging through the result set. Typically, server-driven paging adjusts batch boundaries to accommodate changing data sets more effectively than client-driven paging.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

The `Server Driven Pagination` field on the external data source specifies whether to use client-driven or server-driven paging. If you enable server-driven paging on an external data source, Salesforce ignores the requested page sizes, including the default `queryMore()` batch size of 500 rows. The pages returned by the external system determine the batches. However, the limits for the OData adapters for Salesforce Connect still apply.

SEE ALSO:

- Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter
- General Limits for Salesforce Connect—OData 2.0 and 4.0 Adapters
- OData Query String Options

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

USER PERMISSIONS

To create and edit external data sources:	"Customize Application"
To create and edit external objects:	"Customize Application"
To define or change object-level help:	"Customize Application"
To create and edit custom fields:	"Customize Application"
To edit permission sets and user profiles:	"Manage Profiles and Permission Sets"
To edit another user's authentication settings for external systems:	"Manage Users"

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Let users view and search data that's stored outside your Salesforce org, such as data in an enterprise resource planning (ERP) system.

Setting up Salesforce Connect with anOData 2.0 or 4.0 adapter involves these high-level steps.

1. Define an external data source of type `Salesforce Connect: OData 2.0` or `Salesforce Connect: OData 4.0`.

If your external system hosts multiple services, create an external data source for each service endpoint. Each service endpoint points to an OData service root URL and can expose collections of entities. For example, you'd create a separate external data source for each of these service endpoints.

- `http://services.example.org/Warehouse.svc`
- `https://services.example.org/Payroll.svc`

2. Create the external objects.

   Perform this task only if you don't sync to automatically create the external objects. Create an external object for each external data table that you want to access from your Salesforce org.

3. Create help content for the external objects.

   Create Visualforce pages that describe the external data. When your users click **Help for this Page** on an external object, they read your custom help content. Remember, your users won't find information about the external data in the Salesforce Help.

4. Add custom fields and relationships to the external objects.

   Create relationships between objects. If you didn't sync to automatically create the external objects and their fields, create a custom field for each external table column that you want to access from your Salesforce org.

5. Enable user access to external objects.

   Grant object permissions through permission sets or profiles.

6. Enable user access to the fields on the external objects.

   Grant field permissions through permission sets or profiles.

7. If the external data source uses per-user authentication:

   a. Let users authenticate to the external system.

      Grant users access to authentication settings for the external data source through permission sets or profiles.

   b. Set up each user's authentication settings.

      You or your users can perform this task.

      Tip: Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

SEE ALSO:

OData 2.0 or 4.0 Adapter for Salesforce Connect

Considerations for Salesforce Connect—All Adapters

Developer Guide: *Visualforce Developer Guide*

External Object Relationships

Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

Connect your Salesforce org to data that's stored in an external system, such as SAP® NetWeaver Gateway, Microsoft Dynamics® NAV, or IBM WebSphere®.

Note:

- The external data must be exposed by a service that uses Open Data Protocol (OData) Version 2.0 or 4.0. Such a service is called an *OData producer*.
- The URL for reaching the OData producer must be accessible by Salesforce application servers through the Internet. You can allow access by white-listing Salesforce server IP addresses on your corporate network firewall or by setting up a reverse-proxy XML Gateway.
- The extent to which you can customize data visibility depends on the external system. To determine the optimal settings for integration with Salesforce, consult the external system's documentation.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit external data sources:

- "Customize Application"

Field	Description
Label	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. If you set Identity Type to Per User, this label appears when your users view or edit their authentication settings for external systems.
Name	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Type	Select **Salesforce Connect: OData 2.0** or **Salesforce Connect: OData 4.0**.
URL	The OData service root URL. Make sure that you escape all special characters. Each service endpoint requires its own external data source definition, but you can have multiple entities under one service root URL. For more information about the service root URL and other URL conventions, go to www.odata.org. Examples: • http://services.example.org/Warehouse.svc • https://services.example.org/Payroll.svc If the endpoint is defined in a named credential, enter the named credential URL. A named credential URL contains the scheme callout:, the name of the named credential, and an optional path. For example: callout:*My_Named_Credential/some_path*.

Field	Description
	You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: `callout:My_Named_Credential/some_path?format=json`. If you enter a named credential URL, skip the Authentication section for the external data source. To access the external system, Salesforce Connect uses the authentication settings that are defined in the named credential.
`Connection Timeout`	Number of seconds to wait for a response from the external system before timing out. By default, the value is set to the maximum of 120 seconds. Depending on the availability of and the connection to the external system, it can take a long time to retrieve external data. Use this field to limit how long to wait for external data to load into your org.
`Writable External Objects`	Lets the Force.com platform and users in this org create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.
`High Data Volume`	Salesforce enforces rate limits for retrieving and viewing data from external systems. If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply. See High Data Volume Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters on page 979. High-data-volume external data sources are still limited to 10,000 OData queries per hour for Enterprise, Performance, and Unlimited Editions. Higher limits are available on request. See OData Callout Rate Limit Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters on page 978.
`Server Driven Pagination`	It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. Select this option to have the external system control the paging behavior. See Client-driven and Server-driven Paging for Salesforce Connect—OData 2.0 and 4.0 Adapters on page 970.
`Request Row Counts`	Includes the `$inlinecount=allpages` system query option in OData queries. When selected, the response includes the total row count of the result set. Some external systems don't support the `$inlinecount=allpages` system query option. If you receive errors or notice long load times when you try to access their data, deselect `Request Row Counts` on the external data source. If you do so, however, the external data source and its associated external objects won't support the SOQL `COUNT()` aggregate function, which requires the total row count.
`Compress Requests`	When selected, Salesforce sends compressed HTTP requests to the external system. Make sure that the external system is set up to receive gzip-compressed data. Salesforce automatically accepts gzip-compressed responses.

Field	Description
`Enable Search`	Determines whether SOSL and Salesforce global searches also query the external objects that are associated with this external data source. When selected, you can control which external objects are searchable by selecting or deselecting `Allow Search` on each external object. Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records. Select this option to allow the external data source's associated external objects to appear in the Salesforce1 downloadable app for iOS and the Salesforce1 mobile browser app when used on iOS devices.
`Custom Query Option for Salesforce Search`	Available only for the OData 2.0 adapter for Salesforce Connect. If the OData producer has implemented and exposed a free-text-search custom query option, enter the name of that query string parameter. Learn more about OData custom query options and other URI conventions at `www.odata.org`. This field has no effect when `Enable Search` is deselected or when the OData producer isn't set up to correctly handle the custom query option. See Search Considerations for Salesforce Connect—OData 2.0 Adapter.
`Use Free-Text Search Expressions`	Available only for the OData 4.0 adapter for Salesforce Connect. Select this option to use the `$search` system query option instead of `$filter` in search requests that are sent to the external system. Make sure that the OData producer is set up to support the `$search` system query option. This field has no effect when `Enable Search` is deselected. See Search Considerations for Salesforce Connect—OData 4.0 Adapter.
`Format`	The format that the OData producer uses to represent resources, such as collections of data. Make sure that the OData producer is set up to support the selected format. Learn more about representation formats and operations at `www.odata.org`.
`Special Compatibility`	Select Socrata only if the URL specifies a Socrata open data endpoint. See Socrata™ Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters on page 979.
`CSRF Protection`	If the external system requires Cross-Site Request Forgery (CSRF) protection in requests to create, edit or delete its data, select this option. If you do so, your org obtains an anti-CSRF token and cookie from the external system and includes them in each create, edit, and delete request. See CSRF Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters on page 980. Available only when `Writable External Objects` is selected.
`Anti-CSRF Token Name`	HTTP header field that contains the anti-CSRF token. The external system determines the field name. Default: X-CSRF-Token Available only when `CSRF Protection` is selected.

Field	Description
`Certificate`	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org.
`Identity Type`	Determines whether you're using one set or multiple sets of credentials to access the external system. See Identity Type for External Data Sources on page 934. Select **Anonymous** only if the external system doesn't require authentication.

4. Select the authentication protocol.
   - If you select **Password Authentication**, enter the username and password for accessing the external system.
   - If you select **OAuth 2.0**, complete the following fields.

Field	Description
`Authentication Provider`	Choose the provider. See About External Authentication Providers.
`Scope`	Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Using the Scope Parameter. Note: – The value that you enter replaces the `Default Scopes` value that's defined in the specified authentication provider. – Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. – We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.
`Start Authentication Flow on Save`	To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow. When you click **Save**, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org. Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the `Scope` or `Authentication Provider` fields.

5. Click **Save**.
6. Click **Validate and Sync**, and confirm that the connection is successful.
7. Optionally, select tables and click **Sync** to do the following for each selected table.
   - Automatically create a Salesforce external object.
   - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.

Note: Before you sync, make sure that you understand the considerations that are described in these topics.

- Sync Considerations for Salesforce Connect—All Adapters on page 938
- Sync Considerations for Salesforce Connect—OData 2.0 or 4.0 Adapter on page 941

You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

SEE ALSO:

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

Store Authentication Settings for External Systems

OData Query String Options

OData Type Mapping

Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the special behaviors, limits, and recommendations for using the OData 2.0 or 4.0 adapter for Salesforce Connect.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

OData Producer Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the limits and recommendations for the remote data service that exposes the external data to your Salesforce organization.

OData Callout Rate Limit Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

An external object references data that's stored outside Salesforce. Access to an external object involves a callout to its associated external system. Salesforce enforces rate limits for these callouts.

High Data Volume Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

Socrata™ Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Socrata Open Data Protocol™ is commonly used for health data and for collaboration between governments and their citizens. Salesforce Connect can access data from endpoints that are backed by Socrata Open Data Portal. To accommodate Socrata-specific requirements, set the `Special Compatibility` field on the external data source to Socrata.

CSRF Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the special behaviors, limitations, and recommendations for Cross-Site Request Forgery (CSRF) on OData external data sources.

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

OData Producer Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the limits and recommendations for the remote data service that exposes the external data to your Salesforce organization.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Validate your OData producer by using the Open Data Protocol Service Validation Tool at `services.odata.org/validation`. Doing so checks your implementation against the OData specification and identifies potential issues.
- To improve performance over low-bandwidth connections, set up your OData producer to receive gzip-compressed data. Then, in the external data source definition in Salesforce, select `Compress Requests`. You can also set up the OData producer to send gzip-compressed data to Salesforce, which automatically accepts gzip-compressed responses.
- By default, Salesforce sends each OData request with the `$inlinecount=allpages` system query option. The response then includes the total row count of the result set.

  Some external systems don't support the `$inlinecount=allpages` system query option. If you receive errors or notice long load times when you try to access their data, deselect `Request Row Counts` on the external data source. If you do so, however, the external data source and its associated external objects won't support the SOQL `COUNT()` aggregate function, which requires the total row count. For details about OData URI conventions, go to `www.odata.org`.
- Configure your OData producer to use a page size that's large enough to avoid excessive round trips. Querying a large set of data with a small page size can take a long time because of network latency. Salesforce pages that display external data can take a long time to load.

  For example, if the query results include 100 records, and the page size holds only 5 records, it takes 20 round trips to retrieve the results. If the network latency is 100 ms per round trip, it takes 2 seconds (20 × 100 ms) to retrieve the results.

  In contrast, if the page size holds 20 records, it takes only 5 round trips to retrieve the 100 records. With the same network latency of 100 ms per round trip, it takes 0.5 seconds (5 × 100 ms) to retrieve the results.
- For a list of supported OData types, see "OData Type Mapping" in the Salesforce Help.
- If your external data source uses the OData 4.0 adapter and JSON format, make sure that the OData producer accepts headers that contain the `odata.metadata=full` format parameter. Other variations, including `odata.metadata=minimal`, aren't supported.

SEE ALSO:

OData 2.0 or 4.0 Adapter for Salesforce Connect

OData Query String Options

OData Type Mapping

OData Callout Rate Limit Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

An external object references data that's stored outside Salesforce. Access to an external object involves a callout to its associated external system. Salesforce enforces rate limits for these callouts.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

An org is limited to:

- 10,000 OData callouts per hour for Enterprise, Performance, and Unlimited Editions. Higher limits are available on request.
- 1,000 OData callouts per hour for Developer Edition.

To obtain your org's limits and current usage against those limits, use the `Limits` resource in the Force.com REST API.

Salesforce performs an OData callout each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record. (This doesn't apply to the cross-org adapter.)
- A user runs a report.
- The preview loads in the report builder.
- An external object is queried via flows, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

If your users or applications encounter rate limit errors for OData callouts, try one or more of the following.

- Select `High Data Volume` in the external data source definition. Doing so bypasses most rate limits, but some special behaviors and limitations apply.
- Run fewer SOQL and SOSL queries.
- If you have Apex code that invokes the external system, modify that code to cache frequently accessed external data that seldom changes.
- Contact Salesforce to request a higher limit.

SEE ALSO:

Force.com REST API Developer Guide: List Organization Limits

High Data Volume Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

OData 2.0 or 4.0 Adapter for Salesforce Connect

Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

High Data Volume Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

- The following features aren't available for external objects that are associated with high-data-volume external data sources.
  - Access via Lightning Experience
  - Access via Salesforce1
  - Appearance in Recent Items lists
  - Record feeds
  - Reports and dashboards
  - Writable external objects
- Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.
- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons and links that call JavaScript aren't supported.
- CSRF protection for writable external objects isn't available for high-data-volume external data sources.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Force.com REST API Developer Guide: List Organization Limits

OData Callout Rate Limit Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

CSRF Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

Socrata™ Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Socrata Open Data Protocol™ is commonly used for health data and for collaboration between governments and their citizens. Salesforce Connect can access data from endpoints that are backed by Socrata Open Data Portal. To accommodate Socrata-specific requirements, set the `Special Compatibility` field on the external data source to Socrata.

Socrata doesn't support the row identifier (`_id`) column in `$select` or `$orderby` clauses in OData queries. When Socrata is selected in the `Special Compatibility` field:

- OData queries don't include the `_id` column in `$select` clauses.
- If an `_id` column is synced from a Socrata endpoint, the resulting custom field on the external object isn't sortable.
- If you manually define an external object's custom field with `_id` as the `External Column Name`, make sure that you select the `Sorting Disabled` attribute for that custom field.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

If you modify the `Special Compatibility` field on an external data source, we recommend that you resync its external objects. Or instead, you can test whether queries or user access to the external objects result in errors, and resync only the problematic external objects.

SEE ALSO:

Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

CSRF Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters

Understand the special behaviors, limitations, and recommendations for Cross-Site Request Forgery (CSRF) on OData external data sources.

- CSRF protection isn't available for high-data-volume external data sources.
- Make sure that the URL of the external data source starts with `https://` so that secure HTTP can prevent unauthorized access to the anti-CSRF token and cookie.
- In addition to enabling CSRF protection on the external data source, we recommend keeping CSRF protection enabled in your org's session security settings. These session settings are enabled by default, and keeping them enabled protects your Salesforce data and your external data from CSRF attacks.
  - `Enable CSRF protection on GET requests on non-setup pages`
  - `Enable CSRF protection on POST requests on non-setup pages`

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

SEE ALSO:

Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter

OData Reference for Salesforce Connect—OData 2.0 and 4.0 Adapters

Get to know the Salesforce implementation of the Open Data Protocol (OData) for accessing external systems with Salesforce Connect.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

OData Type Mapping

Salesforce Connect maps OData types to Salesforce metadata field types when syncing metadata and converting values between Salesforce and external systems.

OData Query String Options

The OData adapters for Salesforce Connect use a subset of the OData 2.0 and 4.0 system functions and filter expression constructs to query external systems.

SEE ALSO:

OData 2.0 or 4.0 Adapter for Salesforce Connect

OData Type Mapping

Salesforce Connect maps OData types to Salesforce metadata field types when syncing metadata and converting values between Salesforce and external systems.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

OData 2.0 Type Mapping

Understand how the OData 2.0 adapter for Salesforce Connect maps OData types to Salesforce metadata field types.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Salesforce Connect supports only the following types when syncing metadata and converting values between Salesforce and an external system.

OData 2.0 Primitive Types

OData 2.0 Type	Salesforce Metadata Field Type
Binary	TextArea
Boolean	Checkbox
Byte	Number
DateTime	DateTime
DateTimeOffset	DateTime
Decimal	Number
Double	Number
Guid	Text
Int16	Number
Int32	Number
Int64	Number
SByte	Number
Single	Number
String	Text when the declared length of the OData string column is 255 or fewer characters LongTextArea when the declared length of the OData string column is greater than 255 characters

OData 2.0 Type	Salesforce Metadata Field Type
Time	Text

Tip: A binary value from an external system is represented in Salesforce as a base64-encoded string. You can convert it to a value of type `Blob` by using the `EncodingUtil.base64Decode(inputString)` Apex method.

OData 2.0 Complex Types

Salesforce Connect supports OData complex types as follows.

- External data of complex type is flattened into a string that contains field names and values. For example, an address is flattened into the following string.

```
Street: 55 East 5th Street, City: New York, State: NY, Zip: 10003
```

- An external object custom field associated with an OData complex type on the external system is always read only, even if the external object is writable.

SEE ALSO:

Sync Considerations for Salesforce Connect—All Adapters

OData Reference for Salesforce Connect—OData 2.0 and 4.0 Adapters

Apex Developer Guide: EncodingUtil Class: `base64Decode(inputString)`

OData 4.0 Type Mapping

Understand how the OData 4.0 adapter for Salesforce Connect maps OData types to Salesforce metadata field types.

Salesforce Connect supports only the following types when syncing metadata and converting values between Salesforce and an external system.

OData 4.0 Primitive Types

OData 4.0 Type	Salesforce Metadata Field Type
Binary	TextArea
Boolean	Checkbox
Byte	Number
Date	DateTime
DateTimeOffset	DateTime
Decimal	Number
Double	Number
Guid	Text
Int16	Number
Int32	Number

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

OData 4.0 Type	Salesforce Metadata Field Type
Int64	Number
SByte	Number
Single	Number
String	Text when the declared length of the OData string column is 255 or fewer characters LongTextArea when the declared length of the OData string column is greater than 255 characters

Tip: A binary value from an external system is represented in Salesforce as a base64-encoded string. You can convert it to a value of type `Blob` by using the `EncodingUtil.base64Decode(inputString)` Apex method.

OData 4.0 Complex Types

Salesforce Connect supports OData complex types as follows.

- External data of complex type is flattened into a string that contains field names and values. For example, an address is flattened into the following string.

```
Street: 55 East 5th Street, City: New York, State: NY, Zip: 10003
```

- An external object custom field associated with an OData complex type on the external system is always read only, even if the external object is writable.

SEE ALSO:

Sync Considerations for Salesforce Connect—All Adapters

OData Reference for Salesforce Connect—OData 2.0 and 4.0 Adapters

Apex Developer Guide: EncodingUtil Class: `base64Decode(inputString)`

OData Query String Options

The OData adapters for Salesforce Connect use a subset of the OData 2.0 and 4.0 system functions and filter expression constructs to query external systems.

When your users or the Force.com platform interact with external objects, the OData 2.0 or 4.0 adapter for Salesforce Connect converts those actions into OData queries. Salesforce performs an OData callout each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record. (This doesn't apply to the cross-org adapter.)
- A user runs a report.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The preview loads in the report builder.
- An external object is queried via flows, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

The following sections describe the Salesforce Connect implementation as a consumer of OData services. Salesforce automatically creates the OData queries so that you, as an administrator or developer, don't have to. However, understanding how OData queries are generated—or even attempting manual OData queries—can help you troubleshoot issues with the external system's OData producer. For details about each system query option, go to `www.odata.org`.

Salesforce Connect supports only the following OData system query options. All other options in the OData 2.0 and 4.0 specifications are unused.

`$count` (OData 4.0 only)

Specifies that the response must include the number of rows that the URI identifies after `$filter` system query options are applied, but before `$top` and `$skip` system query options are applied.

When `Request Row Counts` is enabled on the external data source, Salesforce includes `$count=true` in all OData 4.0 queries of that external data source to determine the total number of items in each result set. If `Request Row Counts` is disabled, Salesforce includes `$count=false` in all OData 4.0 queries of the external data source.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$select=SupplierID&` **`$count=true`**`&$top=26`

`$filter`

Filters the collection of resources that's addressed by a request URL. The response contains the results that evaluate to `true`.

Examples	
User action in Salesforce	Open a list view of cities from supplier records that are filtered so that the country is USA.

Examples	
SOQL query	`SELECT City__c FROM Suppliers__x WHERE Country__c = 'USA' ORDER BY City__c ASC LIMIT 26`
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$orderby=City&$select=City,SupplierID&$inlinecount=allpages&$filter=Country+eq+'USA'&$top=26`

`$inlinecount` (OData 2.0 only)

Specifies that the response must include a count of the number of rows that the URI identifies after `$filter` system query options are applied but before `$top` and `$skip` system query options are applied.

When `Request Row Counts` is enabled on the external data source, Salesforce uses `$inlinecount` in all OData 2.0 queries of that external data source to determine the total number of items in each result set. If `Request Row Counts` is disabled, `$inlinecount` is excluded from all OData 2.0 queries of the external data source.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object.
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$select=SupplierID&$inlinecount=allpages&$top=26`

`$orderby`

Sorts the result set in ascending or descending order.

Examples	
User action in Salesforce	Open a list view of supplier records that are ordered by company name.
SOQL query	`SELECT CompanyName__c,ContactName__c FROM Suppliers__x ORDER BY CompanyName__c ASC LIMIT 26`
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$orderby=CompanyName&$select=CompanyName,ContactName,SupplierID&$inlinecount=allpages&$top=26`

`$search` (OData 4.0 only)

Requests entities that match the search query string as a free-text search expression. Enable this option by selecting `Use Free-Text Search Expressions` on the external data source.

By default, `Use Free-Text Search Expressions` is not enabled. The search query string is used as the `contains` value in the `$filter` system query option.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object
Resulting OData query	`http://services.example.org/my.svc/Shippers?$select=CompanyName,Phone,ShipperID&$count=true&`**`$search=Acme`**`&$top=26`

`$select`

Requests a limited set of properties for each entity.

Examples	
User action in Salesforce	Open a list view of supplier records where the page layout displays the company name and contact name.
SOQL query	**`SELECT CompanyName__c,ContactName__c`** `FROM Suppliers__x ORDER BY CompanyName__c ASC LIMIT 26`
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$orderby=CompanyName&`**`$select=CompanyName,ContactName,SupplierID`**`&$inlinecount=allpages&$top=26`

`$skip`

Specifies the number of items in the queried collection to skip in the result set.

Examples	
User action in Salesforce	Click to view the second page of a list view of supplier records that are ordered by city.
SOQL query	`SELECT City__c,CompanyName__c FROM Suppliers__x ORDER BY City__c ASC` **`OFFSET 25`**
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$orderby=City&$select=City,CompanyName,SupplierID&$inlinecount=allpages&$top=26&`**`$skip=25`**

`$top`

Specifies the number of items in the queried collection to include in the result. The value in the `LIMIT` clause of a SOQL query doesn't always match the requested `$top` value, because the latter is modified as needed for client-driven paging and `queryMore()` calls.

Examples	
User action in Salesforce	Open a list view of the top 25 supplier records.
SOQL query	`SELECT SupplierID__c FROM Suppliers__x` **`LIMIT 25`**

Examples	
Resulting OData query	`http://services.example.org/my.svc/Suppliers?$select=SupplierID&$inlinecount=allpages&`**`$top=26`**

OData 2.0 Custom Query Option

If the OData producer supports a custom query string option, enter the name of the query string parameter in the `Custom Query Option for Salesforce Search` field in the external data source definition. Without the custom query option, the search query string is used as the `substringof` value in the `$filter` system query option.

Examples	
OData query without a custom query parameter	`http://services.example.org/my.svc/Shippers?$select=CompanyName,Phone,ShipperID&$inlinecount=allpages&`**`$filter=substringof('Acme',CompanyName)+eq+true+or+substringof('Acme',Phone)+eq+true`**`&$top=26`
OData query with a custom query parameter that's named "doSearch"	`http://services.example.org/my.svc/Shippers?$select=CompanyName,Phone,ShipperID&$inlinecount=allpages&`**`doSearch=Acme`**`&$top=26`

SEE ALSO:

- OData Producer Considerations for Salesforce Connect—OData 2.0 and 4.0 Adapters
- OData Reference for Salesforce Connect—OData 2.0 and 4.0 Adapters
- Client-driven and Server-driven Paging for Salesforce Connect—OData 2.0 and 4.0 Adapters

Salesforce Connect: Custom—Access External Data via Apex

Custom Adapter for Salesforce Connect

Connect to any data anywhere for a complete view of your business. Use the Apex Connector Framework to develop a custom adapter for Salesforce Connect.

Your users and the Force.com platform interact with the external data via external objects. For each of those interactions, Salesforce Connect invokes methods in the Apex classes that compose the custom adapter. Salesforce invokes the custom adapter's Apex code each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record. (This doesn't apply to the cross-org adapter.)
- A user runs a report.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The preview loads in the report builder.
- An external object is queried via flows, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

For details on using the Apex Connector Framework, see "Salesforce Connect" and "DataSource Namespace" in the *Force.com Apex Code Developer's Guide*.

IN THIS SECTION:

External IDs for External Objects in Salesforce Connect—Custom Adapter

When you access external data with a custom adapter for Salesforce Connect, the values of the External ID standard field on an external object come from the `DataSource.Column` named `ExternalId`.

SEE ALSO:

Salesforce Connect

Access External Data with a Custom Adapter for Salesforce Connect

Considerations for Salesforce Connect—Custom Adapter

External IDs for External Objects in Salesforce Connect—Custom Adapter

When you access external data with a custom adapter for Salesforce Connect, the values of the External ID standard field on an external object come from the `DataSource.Column` named `ExternalId`.

Each external object has an `External ID` standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Important:

- The custom adapter's Apex code must declare the `DataSource.Column` named `ExternalId` and provide its values.
- Don't use sensitive data as the values of the External ID standard field, because Salesforce sometimes stores those values.
  - External lookup relationship fields on child records store and display the External ID values of the parent records.
  - For internal use only, Salesforce stores the External ID value of each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.

Example: This excerpt from a sample `DataSource.Connection` class shows the `DataSource.Column` named `ExternalId`.

```
override global List<DataSource.Table> sync() {
    List<DataSource.Table> tables =
    new List<DataSource.Table>();
List<DataSource.Column> columns;
columns = new List<DataSource.Column>();
columns.add(DataSource.Column.text('title', 255));
```

```
columns.add(DataSource.Column.text('description',255));
columns.add(DataSource.Column.text('createdDate',255));
columns.add(DataSource.Column.text('modifiedDate',255));
columns.add(DataSource.Column.url('selfLink'));
columns.add(DataSource.Column.url('DisplayUrl'));
columns.add(DataSource.Column.text('ExternalId',255));
tables.add(DataSource.Table.get('googleDrive','title',
    columns));
return tables;
}
```

SEE ALSO:

Custom Adapter for Salesforce Connect

Apex Developer Guide

Access External Data with a Custom Adapter for Salesforce Connect

USER PERMISSIONS	
To create Apex classes:	"Author Apex"
To configure remote settings:	"Modify All Data"
To create and edit external data sources:	"Customize Application"
To create and edit external objects:	"Customize Application"
To define or change object-level help:	"Customize Application"
To create and edit custom fields:	"Customize Application"
To edit permission sets and user profiles:	"Manage Profiles and Permission Sets"
To edit another user's authentication settings for external systems:	"Manage Users"

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Let users view, search, and modify any data anywhere from within their Salesforce organization.

Setting up Salesforce Connect with a custom adapter involves these high-level steps.

1. Develop the custom adapter for Salesforce Connect.

   Using the Apex Connector Framework, create the `DataSource.Connection` and `DataSource.Provider` classes that comprise the custom adapter.

2. Define remote sites for Apex callouts.

   If the custom adapter involves any Apex callouts, define each callout endpoint as a remote site in your organization. However, you don't need to define a remote site for a callout whose endpoint is specified as a named credential instead of a URL.

3. Define an external data source of type `Salesforce Connect: Custom`.

   If you created multiple custom adapters, make sure that the external data source's `Type` field specifies the correct `DataSource.Provider` class.

4. Create the external objects.

   Perform this task only if you don't sync to automatically create the external objects. Create an external object for each external data table that you want to access from your Salesforce org.

5. Create help content for the external objects.

   Create Visualforce pages that describe the external data. When your users click **Help for this Page** on an external object, they read your custom help content. Remember, your users won't find information about the external data in the Salesforce Help.

6. Add custom fields and relationships to the external objects.

   Create relationships between objects. If you didn't sync to automatically create the external objects and their fields, create a custom field for each external table column that you want to access from your Salesforce org.

7. Enable user access to external objects.

   Grant object permissions through permission sets or profiles.

8. Enable user access to the fields on the external objects.

   Grant field permissions through permission sets or profiles.

9. If the external data source uses per-user authentication:

   a. Let users authenticate to the external system.

      Grant users access to authentication settings for the external data source through permission sets or profiles.

   b. Set up each user's authentication settings.

      You or your users can perform this task.

      Tip: Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

SEE ALSO:

Custom Adapter for Salesforce Connect

Considerations for Salesforce Connect—All Adapters

Developer Guide: *Visualforce Developer's Guide*

External Object Relationships

Define an External Data Source for Salesforce Connect—Custom Adapter

Connect your Salesforce org to any data anywhere via a Salesforce Connect custom adapter that you create with the Apex Connector Framework.

Before you begin, develop the custom adapter for Salesforce Connect. If the custom adapter uses Apex callouts, also define remote sites for the callout endpoints. See Access External Data with a Custom Adapter for Salesforce Connect on page 989.

1. From Setup, enter *External Data Sources* in the `Quick Find` box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
`Label`	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. If you set `Identity Type` to Per User, this label appears when your users view or edit their authentication settings for external systems.
`Name`	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
`Type`	Select **Salesforce Connect: Custom—<your `DataSource.Provider` class name>**.
`URL`	URL of the external system. Available only when the `DataSource.Provider` class declares the `REQUIRE_ENDPOINT` capability. If the class also declares the `REQUIRE_HTTPS` capability, the URL must begin with *`https://`*. If the endpoint is defined in a named credential, enter the named credential URL. A named credential URL contains the scheme `callout:`, the name of the named credential, and an optional path. For example: `callout:`*`My_Named_Credential/some_path`*. You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: `callout:`*`My_Named_Credential/some_path`*`?format=json`.
`Writable External Objects`	Lets the Force.com platform and users in this org create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS

To create and edit external data sources:

- "Customize Application"

Field	Description
`High Data Volume`	Salesforce enforces rate limits for retrieving and viewing data from external systems. If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply. See High Data Volume Considerations for Salesforce Connect—Custom Adapters on page 994.
`Certificate`	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org. Available only when the `DataSource.Provider` class declares the `CERTIFICATE` authentication capability.
`Identity Type`	Determines whether you're using one set or multiple sets of credentials to access the external system. See Identity Type for External Data Sources on page 934. Available only when the `DataSource.Provider` class declares the `BASIC` or `OAUTH` authentication capability.

4. Select the authentication protocol.
   - If you select **Password Authentication**, enter the username and password for accessing the external system.

     Password authentication is available only when the `DataSource.Provider` class declares the `BASIC` authentication capability.
   - If you select **OAuth 2.0**, complete the following fields.

     OAuth 2.0 is available only when the `DataSource.Provider` class declares the `OAUTH` authentication capability.

Field	Description
`Authentication Provider`	Choose the provider. See About External Authentication Providers.
`Scope`	Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Using the Scope Parameter. Note: – The value that you enter replaces the `Default Scopes` value that's defined in the specified authentication provider. – Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. – We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.
`Start Authentication Flow on Save`	To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow. When you click **Save**, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org.

Field	Description
	Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the `Scope` or `Authentication Provider` fields.

5. Click **Save**.
6. Click **Validate and Sync**, and confirm that the connection is successful.

   This step also invokes the `sync()` method on the `DataSource.Connection` class to obtain the list of tables that you can sync to create external objects and their fields.
7. Optionally, select tables and click **Sync** to do the following for each selected table.
   - Automatically create a Salesforce external object.
   - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.

   Note: Before you sync, make sure that you understand the considerations that are described in these topics.
   - Sync Considerations for Salesforce Connect—All Adapters on page 938
   - Sync Considerations for Salesforce Connect—Custom Adapter on page 942

   You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

SEE ALSO:

Access External Data with a Custom Adapter for Salesforce Connect

Store Authentication Settings for External Systems

Considerations for Salesforce Connect—Custom Adapter

Understand the special behaviors, limits, and recommendations for using a custom adapter for Salesforce Connect.

Also review the considerations that apply to all Salesforce Connect adapters.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

IN THIS SECTION:

Apex Considerations for Salesforce Connect—Custom Adapter

Understand the limits and considerations for creating Salesforce Connect custom adapters with the Apex Connector Framework.

High Data Volume Considerations for Salesforce Connect—Custom Adapters

If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

SEE ALSO:

Considerations for Salesforce Connect—All Adapters

Apex Considerations for Salesforce Connect—Custom Adapter

Understand the limits and considerations for creating Salesforce Connect custom adapters with the Apex Connector Framework.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Data type limitations:
  - Double—The value loses precision beyond 18 significant digits. For higher precision, use decimals instead of doubles.
  - String—If the length is greater than 255 characters, the string is mapped to a long text area field in Salesforce.
- Custom adapters for Salesforce Connect are subject to the same limitations as any other Apex code. For example:
  - All Apex governor limits apply.
  - Apex callouts aren't allowed after data manipulation language (DML) operations in the same transaction. Therefore, within the same transaction, you can't update a Salesforce record and then do an Apex callout.
  - Test methods don't support web service callouts; tests that perform web service callouts fail. For an example that shows how to avoid these failing tests by returning mock responses, see Google Drive™ Custom Adapter for Salesforce Connect.

SEE ALSO:

Custom Adapter for Salesforce Connect

Salesforce Compatibility Considerations for Salesforce Connect—All Adapters

Apex Developer Guide: Primitive Data Types

Apex Developer Guide: Execution Governors and Limits

Apex Developer Guide: Callout Limits and Limitations

Apex Developer Guide: Google Drive™ Custom Adapter for Salesforce Connect

High Data Volume Considerations for Salesforce Connect—Custom Adapters

If your org hits rate limits when accessing external objects, consider selecting the `High Data Volume` option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The following features aren't available for external objects that are associated with high-data-volume external data sources.
  - Access via Lightning Experience
  - Access via Salesforce1
  - Appearance in Recent Items lists
  - Record feeds
  - Reports and dashboards
  - Writable external objects
- Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.

- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons and links that call JavaScript aren't supported.

SEE ALSO:

Custom Adapter for Salesforce Connect

Considerations for Salesforce Connect—Custom Adapter

Define External Data Sources

Create external data sources to connect to content and data that is stored outside your Salesforce org.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**. To modify an existing external data source, click **Edit**.
3. Complete the steps for your type of external data source.
   - Files Connect: Google Drive
   - Files Connect: SharePoint 2010 or 2013
   - Files Connect: SharePoint Online or OneDrive for Business
   - Simple URL: Data from Another Web Domain
   - Salesforce Connect: OData 2.0
   - Salesforce Connect: OData 4.0
   - Salesforce Connect: Cross-Org
   - Salesforce Connect: Custom

SEE ALSO:

Salesforce Connect

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and edit an external data source:
- "Customize Application"

Validate and Sync an External Data Source

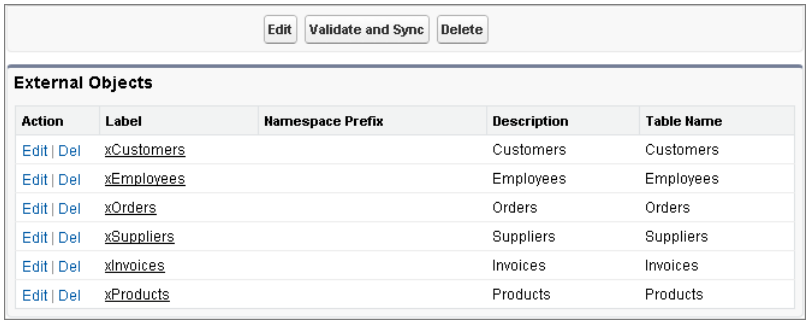
After you configure an external data source, synchronize it to map its tables with external objects in your Salesforce org. The content and data of external objects appear in federated search, together with your Salesforce content and data.

Note:

- Syncing creates or overwrites Salesforce external objects that map to the external system's schema. Syncing doesn't copy any data into your Salesforce org or write data from your org to the external system.
- Syncing is a one-time process. If the external system's schema is changed, the changes aren't automatically synced to your Salesforce org. Resync the objects to reflect the changes in the external system.
- Each org can have up to 100 external objects. Syncing fails if it causes your org to exceed this limit.
- For Salesforce Connect external data sources, make sure that you read and understand the sync considerations. See Sync Considerations for Salesforce Connect—All Adapters on page 938.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click the name of the external data source.
3. Click **Validate and Sync**, and confirm that the connection is successful.
4. Select tables and click **Sync** to do the following for each selected table.
   - Automatically create a Salesforce external object.
   - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.

Example: The resulting external data source detail page will include a list of related external objects like this.



SEE ALSO:

Salesforce Connect

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create an external object from an external data source:

- "Customize Application"

Define External Objects

Tables in external data sources map to external objects in Salesforce, combining all your data and content in federated search.

External objects are similar to custom objects, except that they map to data that's stored outside your Salesforce organization. Each external object relies on an external data source definition to connect with the external system's data. Each external object definition maps to a data table on the external system. Each of the external object's fields maps to a table column on the external system. External objects enable your users and the Force.com platform to search and interact with the external data.

Note:

- Each org can have up to 100 external objects.
- If the external system allows it, we recommend that you sync the external data source to automatically create related external objects. You can instead choose to manually define external objects to customize the external object names and manually create the custom fields.

To create or modify an external object:

1. From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**.
2. Click **New External Object**, or click **Edit** to modify an existing external object.
3. Enter the following:

Field	Description
Label	A user-friendly name for the external object. The label is displayed in the Salesforce user interface, such as in list views. We recommend that you make object labels unique across all standard, custom, and external objects in the org.
Plural Label	The plural name of the external object. If you create a tab for this object, this name is used for the tab.
Starts with vowel sound	If it is appropriate for your organization's default language, select to precede your label with "an" instead of "a" for any automated messages.
Object Name	A unique identifier used to refer to this external object definition when using the API. Object names must be unique across all standard, custom, and external objects in the org. The Object Name field can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Description	An optional description of the external object. A meaningful description helps you distinguish among your external objects when you view them in a list.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create or edit external objects:

- "Customize Application"

Field	Description
`Context-Sensitive Help Setting`	Defines what appears when users click **Help for this Page** from the external object record home (overview) and detail pages, as well as list views and related lists. We recommend that you select **Open a window using a Visualforce page** to display custom help that you create for your users. See Getting Started with Object-Level Help on page 51. If you instead keep the default value, your users only see the Salesforce Help, which doesn't provide any information about your external data. This setting doesn't affect the **Help & Training** link at the top of each page, which always opens the Salesforce Help.
`Content Name`	Select the Visualforce page that best describes the data that's provided by this external object. This field is available only when you select **Open a window using a Visualforce page**.
`External Data Source`	The external data source definition that contains the connection details you want to use for this external object.
`Table Name`	Table in the external system that the external object maps to. For SharePoint, the table name must match the related scope name.
`Display URL Reference Field`	Available only for Salesforce Connect. The external object's `Display URL` standard field values are automatically generated from the external system. For example, with the OData 2.0 adapter for Salesforce Connect, the value is based on the `link href` that's defined on the OData producer. You can override the default values with the values of a custom field on the same external object. Select the field name, and make sure that the custom field's values are valid URLs.
`Allow Reports`	Available only for Salesforce Connect.
`Deployment Status`	Indicates whether the external object is visible to other users.
`Launch New Custom Tab Wizard after saving this external object`	If selected, the custom tab wizard starts after you save the external object.
`Allow Search`	If search is also enabled on the external data source, selecting this option lets users find the external object's records via SOSL and Salesforce global searches. By default, search is disabled for new external objects. However, you can validate and sync an external data source to automatically create external objects. Syncing always enables search on the external object when search is enabled on the external data source, and vice versa. See Search Considerations for Salesforce Connect—All Adapters on page 947.

4. Click **Save**.
5. On the external object detail page, view and modify the external object's custom fields and relationships, page layouts, field sets, search layouts, and buttons and links.

- To create field mappings or add fields to an external object, click **New** on the Custom Fields & Relationships related list.
- To assign different page layouts by user profile, click **Page Layout Assignments**.

SEE ALSO:

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Access External Data with a Custom Adapter for Salesforce Connect

Grant Access to Authentication Settings for External Data Sources

For external data sources that use per-user authentication, grant users access through permission sets and profiles. Doing so lets users set up and manage their own authentication settings for accessing the external system.

1. From Setup, enter *Permission Sets* in the Quick Find box, then select **Permission Sets** or **Profiles**.
2. Click the name of the permission set or profile that you want to modify.
3. Do one of the following.
   - For a permission set, or for a profile in the enhanced profile user interface, click **External Data Source Access** in the Apps section. Then click **Edit**.
   - For a profile in the original profile user interface, click **Edit** in the Enabled External Data Source Access section.
4. Add the data sources that you want to enable.
5. Click **Save**.

SEE ALSO:

Store Authentication Settings for External Systems

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Access External Data with the OData 2.0 or 4.0 Adapter for Salesforce Connect

Access External Data with a Custom Adapter for Salesforce Connect

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit permission sets and user profiles:

- "Manage Profiles and Permission Sets"

Store Authentication Settings for External Systems

You or your administrator can set up and manage your authentication settings for external systems. With valid authentication settings, you can access external systems from within your Salesforce org.

Your administrator defines external systems in external data sources and named credentials. External data sources specify how to access data or content that's stored outside your Salesforce org. Named credentials specify callout endpoints, which receive Web service callouts from your org.

Before you begin, your administrator:

- Sets up the external data source or named credential to use per-user authentication.
- Grants you access to the external data source or named credential.
- Verifies that your org can connect to the external system.
- Tells you the authentication settings to enter.
- Sets up the community, if applicable, by using the Salesforce Tabs + Visualforce template.

  If the community is built with the Customer Service (Napili) template, only your administrator can set up and manage your authentication settings for external systems. You can't complete these steps on your own.

If you don't see the expected settings or options, contact your administrator.

1. Access your authentication settings for external systems in one of the following ways.

   From within a community:

   - If you have a community license, click **My Settings** > **Authentication Settings for External Systems**.
   - Otherwise, from your personal settings, enter *Authentication*, then select **Authentication Settings for External Systems**

   From Salesforce, go to your personal settings and enter *Authentication* in the Quick Find box, then select **Authentication Settings for External Systems**.

2. Click **New** or **Edit**.
3. Complete the fields.

Field	Description
External System Definition	If you're not sure which option to select, ask your administrator. • External Data Source: Provides access to external objects, whose data is stored outside your Salesforce organization. • Named Credential: Enables your actions to trigger authenticated callouts to the endpoint that's specified in the named credential. A named credential can handle the authentication for an external data source. In this scenario, your administrator instructs you to select Named Credential in this field to access external objects.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Named credentials are available in: **All** Editions.

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To store authentication settings for an external data source:

- The data source enabled under "External Data Source Access"

To store authentication settings for a named credential:

- The named credential enabled under "Named Credential Access"

To edit another user's authentication settings for external systems:

- "Manage Users"

Field	Description
`External Data Source` or `Named Credential`	Which field appears depends on what's selected for `External System Definition`. If you're not sure which option to select, ask your administrator. Your administrator can change the option labels to make them more relevant or easier to distinguish from each other.
`User`	Available only to administrators. Select the user whose authentication settings you're entering.

4. Select the authentication protocol that's required by the external system. If you're not sure which option to select, ask your administrator.
   - If you select **Password Authentication**, enter your username and password for the external system.
   - If you select **OAuth 2.0**, complete the following fields.

Field	Description
`Authentication Provider`	If you're not sure which option to select, ask your administrator. Your administrator can change the option labels to make them more relevant or easier to distinguish from each other.
`Scope`	If you're not sure what to enter, ask your administrator. Specifies the scope of permissions to request for the access token.
`Start Authentication Flow on Save`	To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow. When you click **Save**, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org. Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the `Scope` or `Authentication Provider` fields.

5. Click **Save**.

SEE ALSO:

Define External Data Sources

Grant Access to Authentication Settings for External Data Sources

External Object Relationships

External objects support standard lookup relationships, which use the 18-character Salesforce record IDs to associate related records with each other. However, data that's stored outside your Salesforce org often doesn't contain those record IDs. Therefore, two special types of lookup relationships are available for external objects: external lookups and indirect lookups.

External lookups and indirect lookups compare a specific field's values on the parent object to the relationship field's values on the child object. When values match, the records are related to each other.

To create an external object relationship, create a custom field on the child object with one of the following field types. If the child is an external object, you can instead change the field type of an existing custom field to one of the following.

- Lookup Relationship
- External Lookup Relationship
- Indirect Lookup Relationship

This table summarizes the types of relationships that are available to external objects.

Relationship	Allowed Child Objects	Allowed Parent Objects	Parent Field for Matching Records
Lookup	Standard Custom External	Standard Custom	The 18-character Salesforce record ID
External lookup	Standard Custom External	External	The External ID standard field
Indirect lookup	External	Standard Custom	You select a custom field with the `External ID` and `Unique` attributes

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Lookup Relationship Fields on External Objects

Use a lookup relationship when the external data includes a column that identifies related Salesforce records by their 18-character IDs.

External Lookup Relationship Fields on External Objects

Use an external lookup relationship when the parent is an external object.

Indirect Lookup Relationship Fields on External Objects

Use an indirect lookup relationship when the external data doesn't include Salesforce record IDs.

SEE ALSO:

Create Custom Fields

Change the Custom Field Type

Relationship Considerations for Salesforce Connect—All Adapters

Object Relationships Overview

Lookup Relationship Fields on External Objects

Use a lookup relationship when the external data includes a column that identifies related Salesforce records by their 18-character IDs.

A lookup relationship field links a child standard, custom, or external object to a parent standard or custom object. A user who's editing a child record can click the field's lookup icon to select a specific parent record, and a user who's viewing a parent record can view a related list of child records.

When you create a lookup relationship field on an external object, enter the External Column Name that contains the 18-character Salesforce IDs for identifying the parent records.

Example:

- Account record (parent standard object) displays a related list of external SAP sales orders (child external object).
- Account record (parent standard object) displays a related list of support cases (child standard object).

SEE ALSO:

External Object Relationships

Create Custom Fields

Relationship Considerations for Salesforce Connect—All Adapters

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

External Lookup Relationship Fields on External Objects

Use an external lookup relationship when the parent is an external object.

An external lookup relationship links a child standard, custom, or external object to a parent external object.

The values of the standard External ID field on the parent external object are matched against the values of the external lookup relationship field. For a child external object, the values of the external lookup relationship field come from the specified External Column Name.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

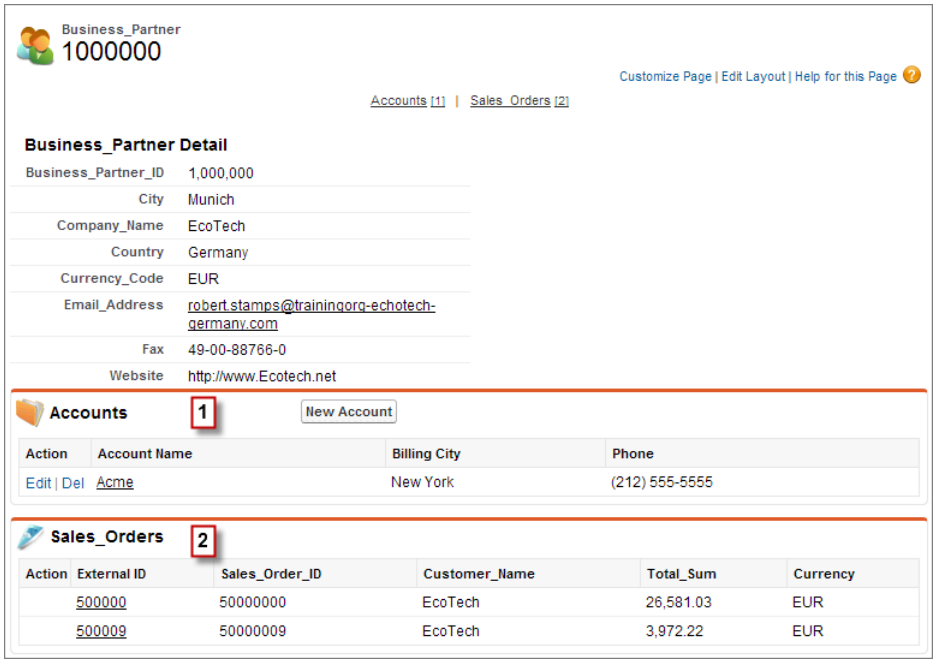
Example:

- External product catalog item (parent external object) displays a related list of support cases (child standard object).
- External customer (parent external object) displays a related list of external orders (child external object).

Example: For the cross-org adapter for Salesforce Connect, suppose that you store contacts and accounts in the provider org. From the subscriber org, you want to view each account's related contacts. To do so, create an external lookup field on the subscriber org's Contact external object. Link that external lookup field to the subscriber org's Account external object. Then set up the page layouts for the Account external object to include a related list that displays the related Contact external object records.

Example: In this screenshot, a record detail page for the Business_Partner external object includes two related lists of child objects. This example shows how external lookup relationships and page layouts enable users to view related data from within and from outside their Salesforce org on a single page.

- Account standard object (1)
- Sales_Order external object (2)



SEE ALSO:

- External Object Relationships
- Create Custom Fields
- Change the Custom Field Type
- Relationship Considerations for Salesforce Connect—All Adapters

Indirect Lookup Relationship Fields on External Objects

Use an indirect lookup relationship when the external data doesn't include Salesforce record IDs.

An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match against each other.

Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values are determined by the specified External Column Name.

If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID**, **Unique**, and **Treat "ABC" and "abc" as different values (case sensitive)**.

Note: Only objects that have a custom field with the `External ID` and `Unique` attributes are available as parent objects in indirect lookup relationships. If you don't see the desired object when you create an indirect lookup relationship field, add a custom unique, external ID field to that object.

Example:

- Account record (parent standard object) displays a related list of SAP sales orders (child external object) with matching customer IDs that aren't Salesforce IDs.
- Contact record (parent standard object) displays a related list of social media posts (child external object) with matching social media handles.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Files Connect for on-premises external data sources is available for an extra cost in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

- External Object Relationships
- Create Custom Fields
- Change the Custom Field Type
- Relationship Considerations for Salesforce Connect—All Adapters

Organization Sync

Getting Started with Organization Sync

With Organization Sync, you can set up a secondary, synced Salesforce organization which users can access whenever your primary organization is experiencing downtime or maintenance.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Organization Sync

At Salesforce, we recognize that you need to access your data at a moment's notice—even during our downtimes and maintenance. With Organization Sync, you can set up a secondary, synced Salesforce organization where users can work on your most business-critical processes and data whenever your primary organization is experiencing downtime or maintenance.

Considerations for Setting Up Organization Sync

Before you start to set up Organization Sync, review this essential information. The way you currently use auto-numbered fields, Apex triggers, audit fields, and other Salesforce features determines how you approach your implementation.

Set Up the Organization Sync Environment

As an Organization Sync administrator, you have a primary and a secondary Salesforce organization, each with a different administrator username and password. Before you can add data to the secondary organization, you must complete a few setup tasks and create a replication connection between the two organizations.

Define the Organization Sync User Experience

Understand the options and limitations behind user rerouting, logins, and data in Organization Sync.

Publish and Subscribe to Objects in Organization Sync

Once the primary and secondary organizations are connected via the Organization Sync connection, it's time to perform a publish and subscribe process that allows record inserts, updates, and deletes to flow between the organizations.

Perform a Bulk Sync

A bulk sync triggers the copying of any new record whose object type is published and subscribed, making the record available in both organizations. A bulk sync copies records in one direction, so performing a bulk sync in the primary organization copies new records to the secondary organization.

Managing Organization Sync

Stay informed about the health of your Organization Sync environment.

Organization Sync

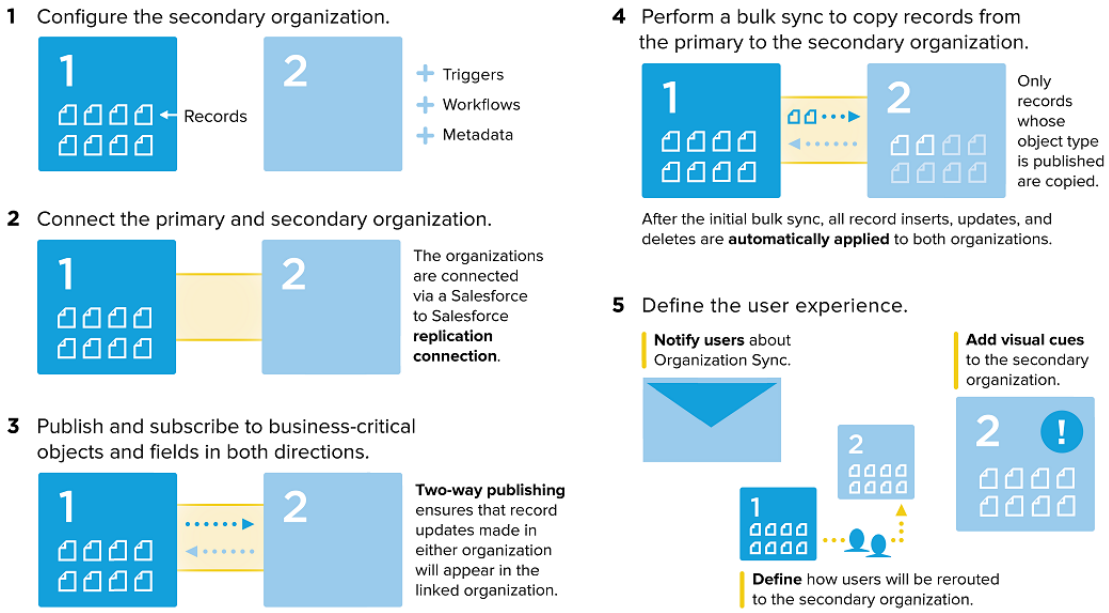
At Salesforce, we recognize that you need to access your data at a moment's notice—even during our downtimes and maintenance. With Organization Sync, you can set up a secondary, synced Salesforce organization where users can work on your most business-critical processes and data whenever your primary organization is experiencing downtime or maintenance.

Organization Sync is ideal for companies whose users need access to Salesforce at all times. For example, representatives in a 24/7 call center can continue to serve customers during maintenance windows with minimal disruption.

Organization Sync uses the framework of Salesforce to Salesforce, a feature that lets customers share records with other companies that use Salesforce. To set up Organization Sync, the Salesforce administrator establishes a secondary Salesforce organization with matching metadata and creates a data replication connection between the primary and secondary organizations. The administrator specifies which types of data can be copied to the secondary organization by mapping essential objects and fields to the secondary organization through a publish and subscribe process. Then, records are copied to the secondary organization in a way that ensures that they remain synced.

When the primary organization is down for maintenance, users are redirected to the secondary organization until the maintenance is complete. All actions (creating a case, deleting an attachment, and more) that users perform in one organization are automatically applied to the other organization. The administrator can also choose to redirect users at any time if he or she needs to make updates to the primary organization.

This graphic shows the steps the Salesforce administrator follows to implement Organization Sync:



Want to learn more about the Organization Sync structure and setup? Watch this short, informative video: How Organization Sync Works.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To set up a replication connection:

- "Manage Connections"

Considerations for Setting Up Organization Sync

Before you start to set up Organization Sync, review this essential information. The way you currently use auto-numbered fields, Apex triggers, audit fields, and other Salesforce features determines how you approach your implementation.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Organization Sync Limitations

Learn about Organization Sync's limitations.

Syncing Feeds and Audit Fields

Syncing feeds and audit fields on connected records makes it easier for users working in the secondary organization to quickly assess a record's history.

Configuring Triggers

If you have Apex triggers in place in your primary Salesforce organization, they aren't automatically duplicated to the secondary organization in Organization Sync. This is because it may not make sense for a trigger to be active in both organizations. However, you can adapt your primary organization triggers to meet your needs in both organizations.

Configuring Auto-Numbered Fields

Fields with auto-generated numbers such as case numbers aren't synced in Organization Sync, but you can develop a workaround to keep the user experience consistent in the primary and secondary organization.

Choosing a Syncing Approach

Though Organization Sync is designed to support a direct two-way sync between the primary and secondary organization, it can also support a one-way sync scenario. Before beginning the publish and subscribe process, decide which approach best meets your business needs.

Organization Sync Limitations

Learn about Organization Sync's limitations.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Organizational Limitations

- To use Organization Sync, you must purchase a secondary Salesforce org and its accompanying user licenses.
- The primary and secondary orgs must be the same organization type (for example, production or sandbox) and edition.
- To avoid inconsistent behavior, both orgs must have the same limits and settings. If the limits of the primary org have been adjusted, contact Salesforce to have the same changes made in the secondary org.
- To avoid Duplicates Detected errors, we recommend turning off duplicate management in secondary orgs.
- Organization Sync can be configured to work for more than two orgs, but it is designed to support a two-org setup.

Unsupported Fields

- Auto-numbered fields aren't synced, but you can implement a workaround. To learn more, see Configuring Auto-Numbered Fields.
- If you've enabled the option to sync feeds and audit fields, audit fields don't appear in the list of publishable fields because they're automatically synced. Audit fields are only synced when a record is copied from one org to another (for example, during a bulk sync). If an audit field on a synced record is updated, that update is not reflected in the linked org.
- Rich text area (RTA) fields aren't synced.

- Formula fields and roll-up summary fields can't be published because their values are determined by other fields' values. As a workaround, make sure that the input fields for a formula or summary roll-up field are published. Then, create an identical formula or roll-up summary field in the secondary organization that refers to those input fields to calculate its value.
- Lookup fields to unpublished objects aren't automatically published because publishing such fields can cause syncing errors. For example, if a field on the Contact object includes a lookup to accounts and the Account object hasn't yet been published, that field can only be published manually.
- Callout fields such as geolocation fields aren't synced.

Unsupported Objects and Features

- Only certain objects can be published in Organization Sync. For a list, see Which Objects Can Be Published?.
- Certain types of metadata aren't supported. For a list, see Which Types of Metadata Are Synced?.
- Supported metadata is only synced in one direction, from the primary to the secondary org.
- LiveAgent, Salesforce Knowledge, Portals, and Communities aren't supported.
- Managed packages aren't synced. If you wish, you can install a package on both organizations to keep the user experience consistent.
- Chatter file posts larger than 2 GB aren't synced.
- Chatter topics, groups, and likes aren't synced.
- In synced Chatter posts, user mentions aren't clickable.
- Chatter file versions aren't supported.

Automation and Data Structure

- Multi-level lookups with more than one level are only synced during bulk syncs.
- Apex triggers, processes, and workflows aren't duplicated from one org to the other because duplicate automation can lead to looping updates or errors. You may need to adapt your existing triggers, processes, and workflows for the secondary org.

Syncing Limitations

- Users are only synced in one direction, from the primary to the secondary org. For more information, see Managing User Data in Organization Sync.
- A bulk sync can only copy records that were last created or modified no more than *one year* before the bulk sync date.
- If the secondary org contained records before you established a connection between the two orgs, those pre-existing records in the secondary org aren't auto-connected to matching records in the primary org. In this situation, bulk syncing entities that already exist in both orgs will cause duplicate alerts if duplicate management is enabled.
- Salesforce does not recommend using a two-way sync for some objects and fields and a one-way sync for others.

Syncing Feeds and Audit Fields

Syncing feeds and audit fields on connected records makes it easier for users working in the secondary organization to quickly assess a record's history.

Records and feed posts in Salesforce contain four audit fields:

- `CreatedBy`
- `CreatedDate`
- `LastModifiedBy`
- `LastModifiedDate`

If you opt to sync audit fields, keep in mind that audit fields are only synced when a record is copied from one org to another (for example, during a bulk sync). If an audit field on a synced record is

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

updated, that update is not reflected in the linked org. We recommend syncing audit fields to give users as consistent an experience as possible.

To sync feeds and audit fields, select the `Sync Feeds and Audit Fields` option when sending your connection invitation. If you're using a two-way sync, selecting this option syncs audit fields in both directions. You can also edit this option from your Organization Sync connection's detail page.

Important:

- If you don't see the `Sync Feeds and Audit Fields` checkbox when sending the connection invitation, the Salesforce admin must enable audit field creation in your organization. To enable this preference, from Setup, enter **User Interface** in the Quick Find box and click **User Interface**. Then, select `Enable "Set Audit Fields upon Record Creation" and "Update Records with Inactive Owners" User Permissions`.
- Enabling this user interface preference also lets users manually edit audit fields in Salesforce. If your business doesn't allow manual editing of audit fields, you may not be able to sync feeds and audit fields in Organization Sync. In that case, you can still set up feed tracking for records in the secondary organization, but that feed only shows updates made in the secondary organization.

Configuring Triggers

If you have Apex triggers in place in your primary Salesforce organization, they aren't automatically duplicated to the secondary organization in Organization Sync. This is because it may not make sense for a trigger to be active in both organizations. However, you can adapt your primary organization triggers to meet your needs in both organizations.

To adapt your current triggers for use in Organization Sync, reference the `Connection Type` and `Replication Role` fields. These fields appear as columns on the Connections page. The `Connection Type` indicates whether the connection is a standard or replication (Organization Sync) connection, while the `Replication Role` indicates whether the organization is the primary or secondary organization.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

All triggers should be configured before you begin the publish and subscribe process. You'll need to decide which trigger-related behaviors you want to see in the secondary organization. Here are some possible approaches:

- If you'd like an existing trigger to work in both organizations, set up an identical trigger in the secondary organization.
- If an existing trigger isn't needed in the secondary organization, just leave it as is; it will continue to function in the primary organization.
- If you'd like to see different behavior in each organization, set up a trigger that references both organizations.

Example: The following trigger changes all case descriptions to "Case in Primary" in the primary organization, and "Case in Secondary" in the secondary organization.

```
trigger CaseTrigger_UpdateDesc on Case (before insert, before update) {
  boolean isPrimary = true;
  for (PartnerNetworkConnection conn : [select ReplicationRole from
PartnerNetworkConnection where ConnectionType = 'Replication' and ConnectionStatus =
'Accepted']) {
    if (conn.ReplicationRole == 'Secondary') {
      isPrimary = false;
    }
  }

  for (Case c : Trigger.new) {
    if (isPrimary) {
      c.Description = 'Case in Primary';
```

```
    } else {
      c.Description = 'Case in Secondary';
    }
  }
}
```

Configuring Auto-Numbered Fields

Fields with auto-generated numbers such as case numbers aren't synced in Organization Sync, but you can develop a workaround to keep the user experience consistent in the primary and secondary organization.

When a case in the primary organization is copied to the secondary organization, Salesforce views it as a new case and assigns a new case number. However, you can set up a custom field configuration that allows cases to display the same identifying number across both organizations. This lets support agents quickly locate a case by number, regardless of which organization they're working in. While the following example relates to the Case Number field, the same approach can be used for any auto-generated number field in Salesforce.

Important: If your company already uses a custom field to represent case numbers, you don't need to complete these steps.

To configure case numbers for Organization Sync:

1. In the primary organization, create a custom text field named "[Your Company] Case Number."
2. Update the case page layout:
   a. Add the custom case number field that you just created.
   b. Remove or hide the system-generated case number field.
3. Apply a trigger that copies the system-generated case number to the custom case number field, but only runs for cases that already existed or are newly created (rather than copied). This prevents the "false" case number assigned during replication from being copied to the custom case number field. Use the following trigger:

```
trigger UpdateCaseNumber on Case (after insert) {
  List<Case> cases = new List<Case>();
  for (Case c : Trigger.new) {
    if (c.CustomCaseNumber__c == null) {
      Case caseToUpdate = new Case(Id=c.Id);
      caseToUpdate.CustomCaseNumber__c = c.CaseNumber;
      cases.add(caseToUpdate);
    }
  }
  database.update(cases);
}
```

4. Repeat the previous steps in the secondary organization.
5. In the secondary organization, adjust the case number display setting so auto-assigned case numbers begin with "2-". This ensures that system-assigned case numbers won't match existing primary organization case numbers. To change this setting:
   a. From the object management settings for cases, go to the fields area.
   b. Click **Edit** next to the Case Number field.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To add fields to a published object:
- "Manage Connections"

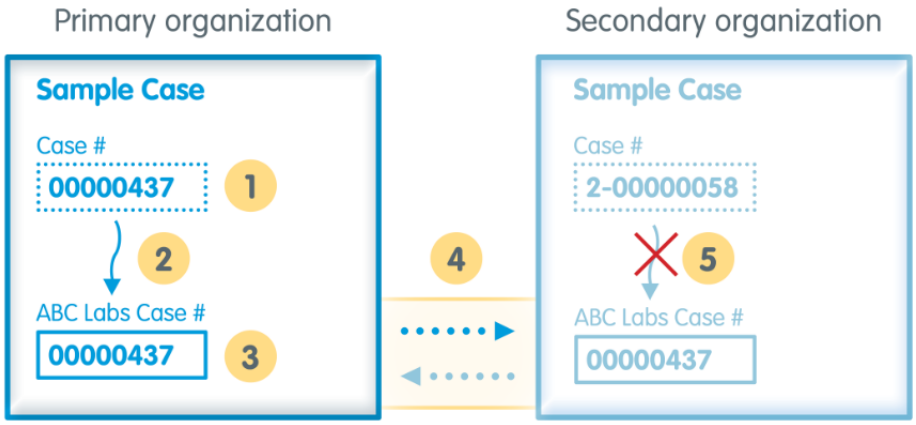
To create or change custom fields:
- "Customize Application"

To create, edit, and delete page layouts:
- "Customize Application"

c. Enter the following text in the Display Format field: **2-{00000000}**.

d. Click **Save**.

The following graphic illustrates the case number configuration. If a two-way sync is implemented, the process remains the same whether the case is created in the primary or the secondary organization.



1: Standard case number field (hidden from case page layout)

2: Trigger

3: Custom case number field

4: Publish and subscribe process

5: Trigger suppressed for replicated cases

Note: For best results, complete these steps before you publish and subscribe to objects and fields; this way, you can publish and subscribe to the custom field you created at the same time as your other business-critical objects and fields. If you have already completed the publish-and-subscribe process, you must publish and subscribe to the custom case number field to make it visible in both organizations.

Choosing a Syncing Approach

Though Organization Sync is designed to support a direct two-way sync between the primary and secondary organization, it can also support a one-way sync scenario. Before beginning the publish and subscribe process, decide which approach best meets your business needs.

If you'd like users to be able to make transactions in both the primary and secondary organizations and have those transactions be reflected in both environments, use a two-way sync. However, if you'd like the secondary organization to merely serve as an up-to-date reference to your primary organization's data, you may prefer a one-way sync for its simplicity. This table provides details about each approach.

Important: Salesforce does not recommend using a two-way sync for some objects and fields and a one-way sync for others. In addition, Organization Sync can be configured to work for more than two organizations, but it is designed to support a two-organization setup.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Details	Two-Way Sync (Standard)	One-Way Sync
How it works	Objects and fields are published from the primary organization and subscribed to in the secondary organization. The same publish and subscribe process is then repeated in the opposite direction (from the secondary to the primary organization) to complete the sync. Records whose object type has been published are then copied to the secondary organization. Updates made to synced records in either organization are automatically applied to the other organization. "Updates" includes creating, updating, deleting, and undeleting records.	Objects and fields are published from the primary organization and subscribed to in the secondary organization. Records whose object type has been published are then copied to the secondary organization. Updates made to synced records in the primary organization are automatically applied to the secondary organization, but not the other way around. Therefore, the secondary organization serves primarily as a reference for users when the primary organization is unavailable.
Things to consider	• Because updates made in the secondary organization are applied to the primary organization, you must account for lookups when setting up Organization Sync to prevent the secondary organization's data from writing over the data in the primary organization. • Triggers, workflows, and Apex code that are in place in the primary organization may need to be modified or turned off in the secondary organization to avoid data loops.	• If users choose to make updates in the secondary organization, they'll need to repeat those updates when they return to working in the primary organization. • When a synced record is undeleted from the secondary organization, its connection with the primary organization is severed. To restore the connection, the record must be manually forwarded from the primary to the secondary organization. • Triggers, workflows, and Apex code in place in the primary organization can also be implemented in the secondary organization, assuming that the objects affected by them are published and subscribed.

Set Up the Organization Sync Environment

As an Organization Sync administrator, you have a primary and a secondary Salesforce organization, each with a different administrator username and password. Before you can add data to the secondary organization, you must complete a few setup tasks and create a replication connection between the two organizations.

Important: Because browsers cache your Salesforce credentials, it's difficult to log in to two different Salesforce organizations in one browser. For best results, work in two browsers. Designate one browser for each organization. For example, use Google Chrome© to access the primary organization and Mozilla Firefox© to access the secondary.

IN THIS SECTION:

1. Enable Salesforce to Salesforce
   Because Organization Sync uses the framework of Salesforce to Salesforce, you need to enable Salesforce to Salesforce in both the primary and secondary organizations. Then, add the Connections tab so you can manage your connection.
2. Duplicate Your Metadata
   Some, but not all, metadata can be synced in Organization Sync. To ensure that data syncs correctly between the two organizations, you must customize and maintain the metadata in your secondary organization.

3. Create a Connection Contact
   To set up the Organization Sync connection, you must assign yourself as the connection contact of the secondary organization. This ensures that you receive emails about the secondary organization, and lets you easily administrate the two organizations.
4. Connect the Organizations
   Once your primary and secondary organizations are configured for Organization Sync, it's time to set up the connection between the primary and secondary organizations. This connection allows updates to flow between the organizations.
5. Explore the Connection Detail Page: Walkthrough
   Take a tour of the Connection Detail page, which is the hub of your Organization Sync connection. Learn how to keep an eye on your connection and update your settings.

Enable Salesforce to Salesforce

Because Organization Sync uses the framework of Salesforce to Salesforce, you need to enable Salesforce to Salesforce in both the primary and secondary organizations. Then, add the Connections tab so you can manage your connection.

Complete these steps in both the primary and secondary organization.

1. Enable Salesforce to Salesforce:
   a. From Setup, enter *Salesforce to Salesforce Settings* in the Quick Find box, then select **Salesforce to Salesforce Settings**.
   b. On the Salesforce to Salesforce page, click **Edit**.
   c. Select **Enable**.
   d. Click **Save**.
2. Add the Connections tab to the user interface:
   a. Click the Plus icon that appears to the right of the current tabs. The All Tabs page appears.
   b. Click **Customize My Tabs**.
   c. Use the **Add** arrow to move the Connections tab from the Available Tabs list to the Selected Tabs list. If you'd like, use the **Up** and **Down** arrows to change the order of the tabs.
   d. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create a Salesforce to Salesforce replication connection:
- "Manage Connections"

Duplicate Your Metadata

Some, but not all, metadata can be synced in Organization Sync. To ensure that data syncs correctly between the two organizations, you must customize and maintain the metadata in your secondary organization.

What is metadata, anyway?

Every time your Salesforce organization is customized, its metadata is modified. For example, editing page layouts, creating a custom field, and creating a record type are all metadata updates. *Data* is information contained in a field, such as a phone number or name. The metadata in your secondary organization must match the metadata in your primary organization. If they don't match, you may not be able to effectively implement Organization Sync.

How do I keep metadata synced between the primary and secondary organizations?

1. Before creating the Organization Sync connection:

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

a. Perform basic data clean-up in the primary organization so your data syncs correctly. Check the primary organization for the following issues, and resolve any that you find:
   - Blank required fields (these can occur if a field was changed from `Optional` to `Required`)
   - Invalid lookups
   - Duplicate records

b. Customize the secondary organization's metadata to match the primary organization's. You can opt to sync some metadata automatically by selecting the **Sync Metadata** checkbox when you send the connection invitation; for a list of supported metadata, see Which Types of Metadata are Synced?. However, if the primary organization has customizations that include unsupported metadata types, you must perform the same customizations or deployments in the secondary organization.

2. After the organizations are connected, develop a business process to ensure that all metadata updates made in one organization are repeated in the other. Remember that if the **Sync Metadata** checkbox is selected, some updates are already synced for you.

When is metadata synced?

If you select the option to sync metadata, Salesforce begins copying supported metadata to the secondary organization as soon as you accept the Organization Sync connection invitation. The `Connection Status` field on your connection's detail page reads `Metadata Sync in Progress` during this process.

After this initial copying of metadata, updates to supported metadata are synced daily. If you need to trigger an immediate sync of all supported metadata, follow these steps:

1. Navigate to your Organization Sync connection's detail page in the primary organization.
2. In the Connection Detail section, click **Edit**.
3. Deselect the **Sync Metadata** checkbox. Click **Save**.
4. In the Connection Detail section, click **Edit** again.
5. Select the **Sync Metadata** checkbox. Click **Save**.

What are best practices for keeping metadata synced?

- If you opt to sync metadata, follow these best practices:
  - If you need to *delete* metadata in one organization, delete it in the other organization as well. This is because metadata deletion isn't synced between the organizations.
  - If you need to create or update *unsupported* metadata, perform the action in both organizations.
  - If you need to create or update *supported* metadata, perform the action in the primary organization only. Because supported metadata is synced daily from the primary to the secondary organization, changes to supported metadata types performed in the secondary organization are overwritten in the daily sync.
  - To keep your metadata current and avoid possible syncing failures, leave the **Sync Metadata** checkbox selected except when triggering an immediate sync as described above. Deselecting the checkbox after a connection invitation is accepted pauses the automatic syncing of metadata updates. The checkbox is editable only from the primary organization.
- If you leave the **Sync Metadata** checkbox deselected, ensure that all metadata changes—creating, deleting, and updating metadata—are repeated in both organizations.

How do integrations and APIs work in Organization Sync?

If you decide that an integration is time-critical to your business operations—for example, an external web form—you must redirect the integration to the secondary organization each time users are redirected there. The same is true for API calls. Any time-critical API calls into Salesforce must be modified to include the secondary organization credentials, or they'll stop processing during the switch to the secondary organization.

Create a Connection Contact

To set up the Organization Sync connection, you must assign yourself as the connection contact of the secondary organization. This ensures that you receive emails about the secondary organization, and lets you easily administrate the two organizations.

On the Connections page, the primary organization is managed by the user in the `Connection Owner` field, while the secondary organization is managed by the user you assign as the `Contact`.

1. In the primary organization, create a contact:
   a. Navigate to the Contacts tab.
   b. Click **New**.
   c. Enter your name in the `First Name` and `Last Name` fields with text that identifies you as the secondary organization contact. For example, if your name is *Elise Medina*, you can enter *Elise Medina (secondary)*.
   d. In the `Account Name` field, use the lookup to select your company's account.
   e. In the `Email` field, enter your own email address.

      Important: This is a critical step that ensures that you'll receive emails about the connection.

   f. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create contacts:
- "Create" on contacts

To view contacts:
- "Read" on contacts

Connect the Organizations

Once your primary and secondary organizations are configured for Organization Sync, it's time to set up the connection between the primary and secondary organizations. This connection allows updates to flow between the organizations.

1. In the primary organization, navigate to the Connections tab.
2. Click **New**.
3. Select the contact name that you created for setting up the connection. The `Account` field will automatically update to list your company.
4. In the `Connection Owner` field, select the administrator you've assigned to manage the connection.
5. In the `Connection Type` field, select `Replication`.
6. If desired, select the option to sync the feeds and audit fields of all records (recommended).

   Note: If you don't see this option, ask your Salesforce admin to enable audit field creation in your organization.

7. If desired, select the option to sync supported metadata (recommended).
8. If desired, select the option to sync users (recommended).
9. Click **Save & Send Invite**.
10. Check your inbox for an email invitation to approve the connection. Click the link in the email to accept the invitation and be redirected to the Salesforce login page.
11. Log in with the secondary organization username and password. You'll see the detail page for your Organization Sync connection. In the Connection Detail section, click **Accept** to complete the connection.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create a Salesforce to Salesforce replication connection:
- "Manage Connections"

Tip: The Connection Detail page is the hub of your Organization Sync connection. After you establish the connection, take a moment to get comfortable with this page via an interactive walkthrough. **Walk Through It**: Explore the Connection Detail Page. Before you begin, make sure to add the Connections tab to your Salesforce environment.

Which Types of Metadata Are Synced?

If you opt to sync metadata by selecting the **Sync Metadata** checkbox on your Organization Sync connection's detail page, certain types of metadata changes are synced daily from the primary to the secondary organization.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Supported Metadata

When the option to sync metadata is selected, *inserts* (adding new metadata) and *updates* (modifying existing metadata) of the following types of metadata are synced daily in one direction—from the primary to the secondary organization.

Supported Metadata	Exceptions
Apex classes, components, and pages	Reports aren't synced in Organization Sync. If an Apex page references a report and you've opted to sync metadata, you must do the following for the Apex page to sync correctly: • Create an identical report in the secondary organization • Update the Apex page to reference the new report ID
Custom fields	
Custom objects	
External objects	
Layouts	Feed item layouts and Salesforce1 settings on layouts aren't synced.
Permission sets	Custom tabs and applications on permission sets aren't synced.
Record types	If an object without any record types was published and subscribed in Organization Sync, and record types are then created for the object, you must re-publish the object for the record types to appear in the secondary organization.
Roles	
User profiles	Custom tabs and applications on user profiles aren't synced.

Unsupported Metadata

The following actions on supported metadata aren't synced.

- Adding and updating metadata in the secondary organization
- Deleting metadata in either organization

For example, if a user in the primary organization creates or updates a permission set, that change is applied to the secondary organization within 24 hours. However, if the permission set is updated in the secondary organization, the update isn't sent back to the primary organization, and the update in the secondary organization will be overwritten in the next sync.

Warning: Renaming supported metadata—for example, renaming a custom field—may break references to that metadata within each organization. To avoid errors, if you need to rename metadata, clone the metadata in question and give the clone your preferred name. Top-level metadata such as roles, profiles, and permission sets can be renamed in the normal way, and the name change is synced as expected.

Explore the Connection Detail Page: Walkthrough

Take a tour of the Connection Detail page, which is the hub of your Organization Sync connection. Learn how to keep an eye on your connection and update your settings.

Greetings, Organization Sync admin!

This is a tour of the Connection Detail page, which is the hub of your Organization Sync connection. There, you can keep an eye on your connection and update your settings.

Walk through it: Explore the Connection Detail Page

1. This walkthrough is for admins who have already set up a replication connection.

   If that's not you, see Configure the Organization Sync Environment for instructions.

2. Click the name of your Organization Sync connection.

   (Hint: It's the connection whose Connection Type is Replication.) This takes you to the Connection Detail page, your one-stop shop for your connection.

3. The Connection Detail section shows general connection settings.

   View the connection status and type, this organization's replication role—primary or secondary—and the contact for each organization.

4. The Published Objects section shows objects that are published to the linked organization.

   When you publish an object, all updates made in this organization to records of that object type are synced with the linked organization.

5. The Subscribed Objects section shows objects that you've subscribed to from the linked organization.

   This section is only visible if you've subscribed to objects. When you subscribe to an object, all updates made in the linked organization to records of that object type are synced with this organization.

6. The Connection History section shows changes to the connection settings.

   These include connection status updates, email notifications, and publishing and subscribing actions.

7. The Record Queue shows record updates that are being sent to the linked organization.

   When a user updates a shared record, the update appears in the queue until the linked organization receives it. The queue only lists outgoing updates, so it's different in each organization.

EDITIONS

Available in:**Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create a replication connection:

- "Manage Connections"

To complete the walkthrough:

- Existing Salesforce to Salesforce replication connection

  AND

  Connections tab added to your list of tabs

8. The Organization Sync Log lists syncing failures.

   If an update can't be sent to the linked organization, it's added to this log so you can investigate further. The full log is available in the Org Sync Log tab.

Way to go!

You've finished your tour of the Connection Detail page. To learn more, check out Organization Sync.

Define the Organization Sync User Experience

Understand the options and limitations behind user rerouting, logins, and data in Organization Sync.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Rerouting Users in Organization Sync
You can define the way users are rerouted to the secondary organization in Organization Sync, and also set up automatic user notifications to let them know when a maintenance period has started.

Managing User Data in Organization Sync
In Organization Sync, user records in the primary organization are mapped to identical user records in the secondary organization. It's important to understand how user-related updates flow between the organizations in case you need to update your configuration.

Managing Secondary Organization Logins
Organization Sync users are given separate login credentials to access the secondary organization. User logins for the secondary organization can be set up and managed just as they are in the primary organization.

How Chatter Works in Organization Sync
While Chatter functions as expected in the secondary organization, some Chatter features aren't synced between the primary and secondary organization.

Rerouting Users in Organization Sync

You can define the way users are rerouted to the secondary organization in Organization Sync, and also set up automatic user notifications to let them know when a maintenance period has started.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Notifying Users About Organization Sync

Since Organization Sync will affect the user experience during maintenance, it's a good idea to notify them about it ahead of time. Let your users know what to expect during a maintenance period, and remind them that they must use different login credentials for the secondary organization. Consider setting up a forum (like a Chatter group) where they can ask questions.

Rerouting Users to the Secondary Organization

When a maintenance window starts, your users will need to be rerouted to the secondary organization URL. Depending on your preference, you can:

- Set up an automatic mass email that tells users to switch to the secondary organization and provides the secondary organization URL.
- Create a URL that reroutes users to the secondary organization during maintenance.
- Set up a Single Sign On solution that reroutes users to the secondary organization during maintenance.

The Salesforce Trust website lists upcoming maintenance windows. You can reroute users to the secondary organization at any time and for any length of time.

Creating Visual Cues

Your users may notice that the secondary organization only contains objects that are supported in Organization Sync. Optionally, you can also create visual cues to let users know that they're working in the secondary organization. For example, add a message to the home page or sidebar.

Managing User Data in Organization Sync

In Organization Sync, user records in the primary organization are mapped to identical user records in the secondary organization. It's important to understand how user-related updates flow between the organizations in case you need to update your configuration.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

How do I add users to the secondary organization?

If you select the **Sync Users** checkbox when sending your Organization Sync connection invitation (recommended), all users with standard Salesforce user licenses are automatically copied to the secondary organization as soon as the connection invitation is accepted. Selecting this option also ensures that user-related updates are synced between your organizations.

If you prefer to add user records to the secondary organization manually, leave the **Sync Users** checkbox deselected when you send the connection invitation. Then, add the user records to the secondary organization using your preferred approach.

Important:

- If you decide to add user records to the secondary organization manually, do so *before* copying your data to the secondary organization via a bulk sync. If a record in the secondary organization references a user that doesn't yet exist in that organization, the field listing the user automatically updates to list the connection owner, and that field value then overwrites the correct value in the primary organization.
- If you've opted to sync users but need to manually trigger the replication of all users, navigate to your Organization Sync connection's detail page in the primary organization. In the Connection Detail section, click **Edit**, deselect the **Sync Users** checkbox, and click **Save**. Then, click **Edit** again, re-select the **Sync Users** checkbox, and click **Save**.

How do users log in to the secondary organization?

Users log in to the secondary organization with a modified version of their primary organization username and a custom password.

How do I know that all users have been added to the secondary organization?

Check the `Connection Status` field on your Organization Sync connection's detail page in the primary organization. As soon as you accept the connection invitation, the connection status reads `Metadata Sync in Progress` if you opted to sync metadata, which signifies that supported metadata is being copied to the secondary organization. The connection status then changes to `User Sync in Progress` while users are copied to the secondary organization. When the connection status changes to `Active`, all user records are available in the secondary organization. User replication failures are listed in the Organization Sync Log.

How long does it take for the user sync to complete?

The length of time required depends on the amount of user data in your primary organization.

Which types of users are synced?

All users with standard Salesforce user licenses are synced. For details about which users fall into this category, see Standard User Licenses.

Are user-related updates synced?

Yes. These updates are synced in real time in both directions between the organizations:

- Adding, updating, or deactivating a user
- Assigning or reassigning a role, user profile, or permission set to a user (as long as the role, user profile, or permission set exists in both organizations)

If you opt to sync metadata, other user-related metadata changes such as creating and updating user profiles are also synced for you.

Note: If you want user updates to only sync in one direction—from the primary to the secondary organization—log in to the secondary organization and deselect the **Sync Users** checkbox on your Organization Sync connection's detail page. This prevents user updates from being sent from the secondary to the primary organization.

What if I choose to sync users but not metadata?

To minimize syncing issues, we recommend selecting both the **Sync Users** and the **Sync Metadata** options when you send the connection invitation. However, if you choose to sync users but not metadata, don't accept the connection invitation until you've updated the metadata in the secondary organization to match the primary organization's metadata. If user-related metadata such as user profiles are missing from the secondary organization, the bulk user sync will fail and will need to be repeated. User syncing failures are listed in the Organization Sync Log.

Managing Secondary Organization Logins

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Organization Sync users are given separate login credentials to access the secondary organization. User logins for the secondary organization can be set up and managed just as they are in the primary organization.

Users log in to the secondary organization with a modified version of their primary organization username and a custom password. Secondary organization usernames end with **.dr**, which ensures that user records map correctly between the two organizations. For example, if a user's username in the primary organization is *tmiller@mycompany.com*, their username in the secondary organization is *tmiller@mycompany.com.dr*.

The **.dr** suffix is automatically added to usernames when a user record is copied to the secondary organization. However, if a new user is created within the secondary organization, the person creating the user must add **.dr** to the username. If **.dr** is not added, the user record won't be copied to the primary organization. The Organization Sync Log in the secondary organization lists username errors.

Note: If a user is locked out of the primary organization after exceeding the maximum number of password attempts, the user can still log in to the secondary organization.

Setting Up User Logins

After the user sync is complete, the Salesforce administrator must log in to the secondary organization and reset all user passwords. This sends an email to all users with a temporary password. Let your users know that this email provides their secondary organization password, which is used with their secondary organization username (their regular Salesforce username with **.dr** appended).

Users can log in to either organization from the Salesforce login page using their corresponding login credentials. Manage user logins for the secondary organization just as you would for the primary organization.

Temporary passwords for new users expire in six months, and users must change their password the first time they log in. The login link in the email can only be used once. A user who follows the link without setting a password must have an administrator reset their password before they can log in.

How Chatter Works in Organization Sync

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

While Chatter functions as expected in the secondary organization, some Chatter features aren't synced between the primary and secondary organization.

If you've enabled a two-way sync, here's what to expect from Chatter in Organization Sync:

Posts

Text, link, and file posts and comments are synced between the organizations, so a post made in one organization is also visible in the other. File posts up to 2 GB are supported.

Note: The organizations must be configured to support synced feeds.

Likes

Users can like posts and comments in the secondary organization just as they would in the primary. However, likes given in one organization aren't visible in the other. Users still receive email notifications about activity on posts or comments that they liked, regardless of whether they liked the post in the primary or the secondary organization. For example, suppose Rita is working in the secondary organization and likes a post. When she returns to working in the primary organization, her like isn't visible on that post. However, if someone in either organization comments on the post she liked, she'll still receive an email notification.

Shares

Users can click **Share** on posts in the secondary organization just as they would in the primary.

Groups

Users can create and edit Chatter groups in the secondary organization just as they would in the primary, but groups aren't synced between the two organizations. This means a group that exists in the primary organization isn't visible in the secondary organization, and vice versa.

Mentions

Users can mention other users and groups in the secondary organization just as they would in the primary. However, mentions made in one organization aren't clickable in the other. For example, suppose a user makes the following comment in the primary organization: "@Ella Roberts, you may be interested in this." In the secondary organization, the comment is visible, but "@Ella Roberts" won't link to Ella's profile.

Topics

Users can add topics to posts in the secondary organization just as they would in the primary, but topics aren't synced between the two organizations. This means a post with topics added to it in the primary organization won't list any topics in the secondary organization.

Publish and Subscribe to Objects in Organization Sync

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Once the primary and secondary organizations are connected via the Organization Sync connection, it's time to perform a publish and subscribe process that allows record inserts, updates, and deletes to flow between the organizations.

In the publish and subscribe process, objects and their fields are published to the primary organization and subscribed to in the secondary organization. Then, to ensure that record updates flow both directions, the same process is repeated from the secondary to the primary organization.

Publishing an object doesn't copy any records to the secondary organization; it simply creates a path where you can then perform a bulk sync to copy records of that object type from the primary to the secondary organization. For example, once you publish the Account object, you can perform a bulk sync to copy all accounts to the secondary organization. After the initial bulk sync, all record inserts, updates, and deletes of that object type are automatically synced across both organizations.

Tip: The Connection Detail page is the hub of your Organization Sync connection. After you establish your connection, take a moment to get comfortable with this page via an interactive walkthrough. **Walk Through It**: Explore the Connection Detail Page. Before you begin, make sure to add the Connections tab to your Salesforce environment.

IN THIS SECTION:

Which Objects Can Be Published?

Only certain objects can be published in Organization Sync. Think carefully about which objects you want to publish. Focus on publishing business-critical objects.

Understanding Lookup Relationships

Before you begin the publish-and-subscribe process, make sure that you understand the role that lookups play in Organization Sync. Organization Sync automatically preserves most relationships for standard lookup relationships. However, you must publish objects and their fields in a way that reflects the hierarchy of your data model.

Publish Objects to the Secondary Organization

Once you've set up the Organization Sync connection, you can publish objects of your choice to the secondary organization.

Edit Published Fields

When you publish an object in Organization Sync, you can also control which fields you publish on that object. All supported fields are published by default when you publish their parent object, but you can unpublish fields if you wish. If a field isn't published, it won't appear in the linked organization.

Subscribe to Objects in the Secondary Organization

Once you publish an object in Organization Sync, you must subscribe to it in the secondary organization. When you subscribe to an object, you map the object to itself to ensure that records of that object type appear in the secondary organization.

Edit Subscribed Fields

When you subscribe to an object in Organization Sync, its fields are automatically subscribed to by default. Fields are auto-mapped to themselves to ensure that your data is synced.

Complete the Publishing Process

Publishing objects in the primary organization and subscribing to them in the secondary organization allows record updates for those object types to be sent from the primary to the secondary organization. If you want updates to be sent both directions, you must publish and subscribe to the same objects in the opposite direction—from the secondary to the primary organization.

Which Objects Can Be Published?

Only certain objects can be published in Organization Sync. Think carefully about which objects you want to publish. Focus on publishing business-critical objects.

When deciding which objects to publish, consider the relationships between objects. If one object is a child of another (for example, Contact is a child of Account), publish both objects to ensure that Contact's lookup to Account functions in the secondary organization.

You can publish the following objects in Organization Sync.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Any custom object that has been deployed

Account

Attachment (Private and Public)

Case

Case Comment (Private and Public)

Case Team Member
This includes predefined case team members, but only applies to case team members that are users. Case team members that are contacts aren't supported.

Contact
If person accounts are enabled in your organization, publish Contact and Account together for best results.

Email Message
Email drafts and inline images aren't supported.

Feed Comment

Feed Item
If you select the option to Sync Feeds and Audit Fields when you establish the Organization Sync connection, the Feed Item and Feed Comment objects must still be published for feeds to be synced between the two organizations. Only text, link, and file posts under 50 MB are supported.

Lead

Opportunity
For best results, update opportunity stages from the opportunity's record detail page. Global find and replace actions for opportunity stages aren't supported in Organization Sync, and won't be synced between the primary and secondary organization.

Product

Task

Understanding Lookup Relationships

Before you begin the publish-and-subscribe process, make sure that you understand the role that lookups play in Organization Sync. Organization Sync automatically preserves most relationships for standard lookup relationships. However, you must publish objects and their fields in a way that reflects the hierarchy of your data model.

Salesforce preserves hierarchical lookup relationships for one levels of lookups. For example, the lookups between the child object Contact and the parent object Account would be preserved.

When a parent entity is deleted in Salesforce, its child entities (such as attachments) are deleted as well. This remains true for Organization Sync. For example, if a case with several attachments is synced, soft deleting the case in either organization causes the case and its attachments to be soft deleted in both organizations. Similarly, undeleting the case in either organization restores the case and its attachments in both organizations.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

To make sure that the lookup relationships of entities being published in the Organization Sync connection stay intact, follow these guidelines:

- Both entities must be selected for publishing and subscribing.
- Publish the parent entity to the secondary organization before or at the same time as the child entity. If the child entity in a lookup relationship is published first, the lookup field is blank; if that lookup field is required, the record replication will fail. For best results, publish all objects together to keep lookups intact.

- There must be no circular dependencies in the data model.

Publish Objects to the Secondary Organization

Once you've set up the Organization Sync connection, you can publish objects of your choice to the secondary organization.

When you publish and subscribe to an object, you create a link between the primary and secondary organization that allows record inserts, updates, and deletes for that object to be synced across both organizations.

To publish objects from the primary to the secondary organization:

1. Log in to the primary organization.
2. Navigate to the Connections tab.
3. In the Connection Name column, click the name of the Organization Sync connection.
4. In the Published Objects section, click **Publish/Unpublish**.
5. Select all of the objects that you want to publish to the secondary organization.

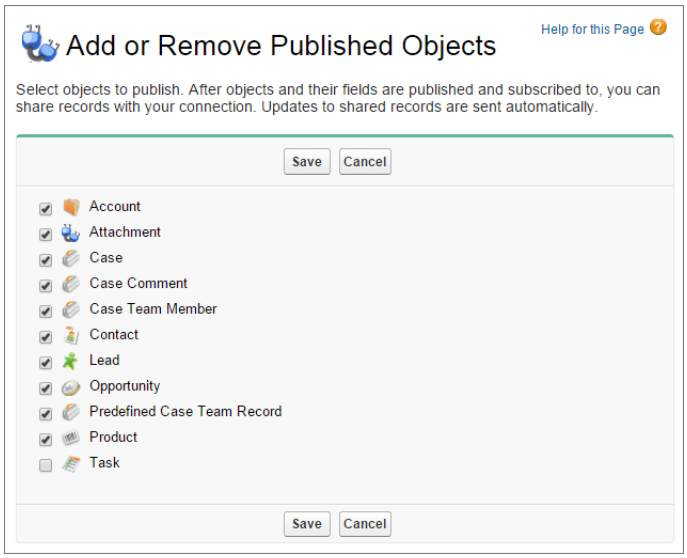
EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit published objects:
- "Manage Connections"



6. Click **Save**. The objects you published appear in the Published Objects section.

Edit Published Fields

When you publish an object in Organization Sync, you can also control which fields you publish on that object. All supported fields are published by default when you publish their parent object, but you can unpublish fields if you wish. If a field isn't published, it won't appear in the linked organization.

To edit an object's published fields:

1. Navigate to the Connections tab.
2. In the Connection Name column, click the name of the Organization Sync connection.
3. In the Published Objects section, click **Edit** next to an object's name.

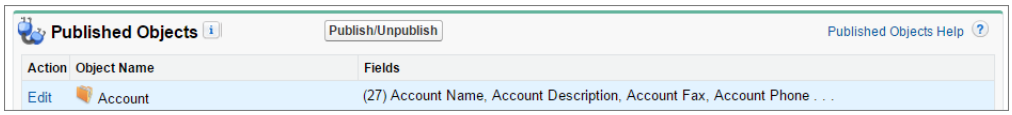
EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit published fields:
- "Manage Connections"



The list of fields related to that object appears. All supported fields are selected by default.

Note:
- Audit fields don't appear in the list of publishable fields because they're automatically synced.
- Rich text area (RTA) fields can't be published.
- Formula fields and roll-up summary fields can't be published because their values are determined by other fields' values. As a workaround, make sure that the input fields for a formula or summary roll-up field are published. Then, create an identical formula or roll-up summary field in the secondary organization that refers to those input fields to calculate its value.
- Lookup fields to unpublished objects aren't automatically published because publishing such fields can cause syncing errors. For example, if a field on the Contact object includes a lookup to accounts and the Account object hasn't yet been published, that field can only be published manually.

4. Deselect any fields you do not want to publish. Before publishing a field, consider which lookup relationships will be affected.
5. Click **Save**.
6. If a two-way sync is implemented and you are unpublishing the field after it was already published and subscribed to, repeat the previous steps in the linked organization.

Warning: When you perform any of the following actions to a field in one organization, the field is automatically unsubscribed in the linked organization:
- Delete the field
- Change the field type
- Change the size or precision of a long text area, text, percent, number, or currency field

The field stays published, but you must re-subscribe to it in the linked organization.

Note: When you need to add a field to an object that is already published in Organization Sync, publish and subscribe to the field to ensure that updates to it are synced. Don't update the field value in existing records until you publish and subscribe to the field.

Subscribe to Objects in the Secondary Organization

Once you publish an object in Organization Sync, you must subscribe to it in the secondary organization. When you subscribe to an object, you map the object to itself to ensure that records of that object type appear in the secondary organization.

To subscribe to objects in the secondary organization:

1. Log in to the secondary organization.
2. Navigate to the Connections tab.
3. In the Connection Name column, click the name of the Organization Sync connection.
4. In the Subscribed Objects section, click **Subscribe/Unsubscribe**.
5. Select **Auto-Map Objects** to map each object to itself. For example, `Account` would map to `Account`.
6. Click **Save**. The objects you subscribed to now appear in the Subscribed Objects section.

When you subscribe to an object in Organization Sync, all of its fields are automatically mapped and subscribed to. If you don't want a particular field to be synced, unpublish it.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit subscribed objects:
- "Manage Connections"

Edit Subscribed Fields

When you subscribe to an object in Organization Sync, its fields are automatically subscribed to by default. Fields are auto-mapped to themselves to ensure that your data is synced.

To view the subscribed fields on an object:

1. Log in to the subscribing organization.
2. Navigate to the Connections tab.
3. Click the name of the Organization Sync connection.
4. In the Subscribed Objects section, click **Edit** next to the object's name.
   A field is subscribed to if the Your Fields column contains a mapping value other than `--None--`. Fields should be mapped to themselves (for example, the Account Description field maps to Account Description).

Note:

- If the Your Fields column reads **None Available** for a field, this means the field is a lookup to an object that hasn't yet been subscribed to. After the object in question is subscribed to, fields that look up to it are automatically subscribed to as well.
- If you don't want a particular field to be synced, just unpublish it; there's no need to unsubscribe to it.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit subscribed fields:
- "Manage Connections"

Complete the Publishing Process

Publishing objects in the primary organization and subscribing to them in the secondary organization allows record updates for those object types to be sent from the primary to the secondary organization. If you want updates to be sent both directions, you must publish and subscribe to the same objects in the opposite direction—from the secondary to the primary organization.

Important: If you are implementing a one-way sync, skip this process.

1. In the secondary organization, publish all objects to the primary organization.
2. In the primary organization:
   a. Subscribe to and map all published objects.
   b. If needed, edit the objects' subscribed fields.

Once you have published and subscribed to objects in both directions—from the primary to the secondary organization, and from the secondary to the primary—you can perform a bulk sync to copy records to the secondary organization. After the initial bulk sync, all record inserts, updates, and deletes are automatically synced across both organizations.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To publish and subscribe to objects:
- "Manage Connections"

Perform a Bulk Sync

A bulk sync triggers the copying of any new record whose object type is published and subscribed, making the record available in both organizations. A bulk sync copies records in one direction, so performing a bulk sync in the primary organization copies new records to the secondary organization.

To perform a bulk sync:

1. Click **Bulk Sync** on your Organization Sync connection's detail page.
   The bulk sync page appears with a list of all published objects.
2. Specify a number of entities to sync, or choose a cutoff date within the past year to control which records are synced.

   Important: The cutoff date defaults to one year before the current date. For the initial bulk sync, consider syncing your records in groups based on date to minimize the amount of data being processed at once.

   For example, you might first perform a bulk sync with a cutoff date of three months ago, and then perform a second bulk sync with a start date of six months ago. Grouping records doesn't take extra time because the system doesn't re-sync records that have already been synced.
3. Click **Start Sync**.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To perform a bulk sync:
- "Manage Connections"

IN THIS SECTION:

Bulk Sync Guidelines

A bulk sync triggers the copying of any record whose object type is published and subscribed, making the record available in both organizations. Once a record is copied in a bulk sync, any future updates to it are synced.

Manually Forward Records

Though you'll primarily use the bulk sync function to copy records to the secondary organization in Organization Sync, it's worth noting that you can also forward individual records from the record detail page in Salesforce.

Unpublish Objects and Fields

To stop an object or field from being synced, you can unpublish it from your organizations in Organization Sync. Be aware of any lookup relationships that are associated with the objects or fields that you unpublish.

Bulk Sync Guidelines

A bulk sync triggers the copying of any record whose object type is published and subscribed, making the record available in both organizations. Once a record is copied in a bulk sync, any future updates to it are synced.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

When should I perform a bulk sync?

When you complete the publish-and-subscribe process as part of the Organization Sync setup, the secondary organization doesn't yet contain any records. Perform an initial bulk sync to quickly populate the secondary organization with records so users can access their data during downtime. In addition, perform a bulk sync whenever you publish and subscribe to new objects.

How do I select a cutoff date on the Bulk Sync page?

A bulk sync triggers the copying of any records that were created or modified after a certain date within the last year. Select this date on the Bulk Sync page. For the initial bulk sync, consider syncing your records in groups based on date to minimize the amount of data being processed at once.

For example, you might first perform a bulk sync with a cutoff date of three months ago, and then perform a second bulk sync with a start date of six months ago. Grouping records doesn't take additional time because the system doesn't recopy records that have already been synced.

How do I monitor the progress of a bulk sync?

View the total number of records remaining to be synced in the Organization Sync Record Queue on your Organization Sync connection's detail page. While the sync is in progress, the **Bulk Sync** button is disabled. Logging out of Salesforce won't affect the sync, and when the sync completes, the **Bulk Sync** button is clickable again.

Can a bulk sync fail?

If a record included in a bulk sync has an issue—for instance, an invalid lookup—that record and its child records are not copied, but records that are unrelated to failed records will still be copied in the bulk sync. Replication failures are logged in the Organization Sync Log, while metadata-related failures are logged in the Connection History log.

How long does a bulk sync take?

The length of a bulk sync depends on the number of records being copied. A bulk sync may take hours or days.

Manually Forward Records

Though you'll primarily use the bulk sync function to copy records to the secondary organization in Organization Sync, it's worth noting that you can also forward individual records from the record detail page in Salesforce.

First, add the External Sharing related list to the page layout of each object type that is published in Organization Sync. The External Sharing related list shows you the sharing status of each record and the connections that the record has been shared with.

1. To forward a record and its related records to the secondary organization:
   a. Navigate to the record's detail page.
   b. In the External Sharing Related List, click **Forward this [object type]**.
2. To forward multiple records to the secondary organization:
   a. Select the tab of the type of record you want to forward.
   b. Select a list view from the `View` drop-down list, then click **Go!**
   c. In the list view, select the records you want to forward. To select all records, select the checkbox in the column header.
   d. Click **Forward to Connections**.
3. Select the Organization Sync connection in the Available Connections list.
4. Click the **Add** arrow to move the connection to the Selected Connections list.
5. Optionally, select related records you want to forward.

   Note: When you forward a case to the secondary organization, you won't see the option to forward its user case team members, feed items, feed comments, or predefined case team because those items are automatically forwarded with the case. Contact case team members are not supported.
6. Click **Save**.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To see the external sharing status of a record:
- "Read" on the record

To manage the Organization Sync connection:
- "Manage Connections"

Unpublish Objects and Fields

To stop an object or field from being synced, you can unpublish it from your organizations in Organization Sync. Be aware of any lookup relationships that are associated with the objects or fields that you unpublish.

When you unpublish an **object** in Organization Sync, any synced records of that object type are disconnected between the two organizations. For example, if you unpublish the Account object and then edit a synced account record in the primary organization, that edit will not carry over to the secondary organization because accounts are no longer synced.

When you unpublish an object's **field** in Organization Sync, synced records of that object type will no longer sync that field. For example, if you unpublish the `Amount` field from the Opportunity object, synced opportunity records will stay synced, but any edits made to the `Amount` field of an opportunity record will not carry over to the other organization. If an opportunity record is created after the `Amount` field is unpublished, it will still be automatically forwarded to the secondary organization, but it won't display the `Amount` field in the secondary organization.

1. To unpublish an object:
   a. Log in to the primary organization.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit published objects:
- "Manage Connections"

b. Navigate to the Connections tab.
c. In the Connection Name column, click on the Organization Sync connection.
d. In the Published Objects section, click **Publish/Unpublish**.
e. Deselect the object you want to unpublish.
f. Click **Save**.
g. If a two-way sync is implemented, complete the previous steps in the secondary organization.

2. To unpublish a field:
   a. Log in to the primary organization.
   b. Navigate to the Connections tab.
   c. In the Connection Name column, click on the Organization Sync connection.
   d. In the Published Objects section, click **Edit** next to the name of the object whose field you want to unpublish.
   e. Deselect the field you want to unpublish.
   f. Click **Save**.
   g. If a two-way sync is implemented, complete the previous steps in the secondary organization.

Managing Organization Sync

Stay informed about the health of your Organization Sync environment.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Monitor the Organization Sync Environment

The connection detail page includes important information that can keep you up-to-date on the status of your Organization Sync connection.

Actions in the Organization Sync Record Queue

When a record update is being sent between the primary and secondary organization, you'll see the record in the Organization Sync record queue on your connection detail page. The `Action` column in the queue lists the replication action that is taking place for a record.

Track Syncing Failures with an Organization Sync Log Trigger

You can write Apex triggers for the Organization Sync Log to stay informed about syncing progress and keep track of failures.

Reporting on the Organization Sync Log

As an Organization Sync administrator, you can create custom report types for the Organization Sync Log to stay informed about syncing failures.

Diagnosing Syncing Issues

Organization Sync users may encounter issues when working between the primary and secondary organization. To pinpoint the cause and resolve the problem, follow the troubleshooting steps.

Suspend Your Organization Sync Connection

If you're experiencing repeated syncing failures or need to make significant changes in your organization, you can suspend the connection between your primary and secondary organizations. Updates made while the connection is suspended aren't synced until you resume the connection.

Monitor the Organization Sync Environment

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The connection detail page includes important information that can keep you up-to-date on the status of your Organization Sync connection.

When a record update is being sent between the organizations, the update appears in the Organization Sync Record Queue on your Organization Sync connection's detail page until the linked organization receives it. The queue shows outgoing syncing actions, so if an update is in the progress of being sent to the secondary organization, it appears in the primary organization queue.

If an update fails to be synced, the system retries it. The `Next Retry Date` column in the queue shows when the system will try again. After multiple retries, the update is logged in the Organization Sync Log, and no longer appears in the queue. The log displays one entry when an update is retried for the first time, and another when the update either fails in the final retry or succeeds. (The user sync and metadata sync don't go through the retry process; if they fail, they'll appear immediately in the Organization Sync Log.)

The Organization Sync Log section on the connection detail page displays the most recent log entries and a link to the full log, which is located in the Org Sync Log tab in Salesforce. Each log entry lists the affected record, the type of update, the time it was logged, the status, and (for errors) an error code. The `Status` field displays `Retrying` when an update fails to sync upon the first attempt, and either `Failed` or `Resolved` after it has been retried multiple times.

Tip:

- If you don't see the event status, add the `Status` field to the Organization Sync Log page layout.
- Click the event number to view the event details and (for errors) a description of the error.

Replication failures are commonly caused by unresolved lookup relationships. If the issue is fixed, the record is re-added to the record queue the next time it is updated. The `Action` column in the queue can offer insight into which operations are failing.

Actions in the Organization Sync Record Queue

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

When a record update is being sent between the primary and secondary organization, you'll see the record in the Organization Sync record queue on your connection detail page. The `Action` column in the queue lists the replication action that is taking place for a record.

Records in the queue will list one of the following actions.

Action	Definition
`Bulk Sync`	Records created or updated after the specified bulk sync date are being copied to the linked organization.
`Bulk Sync Users`	User records are being copied to the linked organization.
`Connection Admin Resumed`	The Organization Sync connection was resumed by a user.
`Connection Admin Suspended`	The Organization Sync connection was suspended by a user. Updates are added to the Organization Sync Record Queue and processed when a user resumes the connection.

Action	Definition
`Connection Deactivate`	The Organization Sync connection was deactivated.
`Connection Resumed`	The Organization Sync connection was automatically resumed because both organizations are now available.
`Connection Suspended`	The Organization Sync connection was automatically suspended because one of the linked organizations is unavailable. Updates are added to the Organization Sync Record Queue and processed when the connection is resumed.
`Connection Update`	The Organization Sync connection owner was updated, and that update is now being sent to the linked organization.
`Field Map Delete`	A field was unpublished, and is now being unpublished in the linked organization.
`Field Map Name Update`	A field name was updated, and that update is now being sent to the linked organization.
`Field Map Update`	A field definition was updated for this record, and that update is now being sent to the linked organization.
`Generic Insert`	The record is being copied to the linked organization.
`Generic Update`	The record was updated, and that update is now being sent to the linked organization.
`Map Update Email`	The mapping of a type, field, or value was updated, and the Salesforce administrator is now being notified by email.
`Missing Partner ID`	The Partner Entity ID for the record was missing, and is being located (no action needed).
`Replication Delete`	The record was deleted, and is now being deleted in the linked organization.
`Replication Undelete`	The record was undeleted, and is now being undeleted in the linked organization.
`Type Map Delete`	A type mapping (such as a custom object definition) was deleted, and is now being deleted in the linked organization.
`Type Map Name Update`	A type mapping name was updated, and that update is now being sent to the linked organization.
`Type Map Set Partner Sub`	A published object was subscribed to, and that update is now being sent to the linked organization.
`Type Map Update`	A type mapping was updated, and that update is now being sent to the linked organization.
`User Upsert`	A user is being copied to the linked organization.
`Value Map Delete`	A picklist value mapping was deleted, and is now being deleted in the linked organization.

Track Syncing Failures with an Organization Sync Log Trigger

You can write Apex triggers for the Organization Sync Log to stay informed about syncing progress and keep track of failures.

Here are some ways you might use triggers for the Organization Sync Log:

- Write a trigger that emails the administrator if more than 10 new entries appeared in the log in the past 24 hours.
- If an entry appears in the log for an account, trigger a workflow rule that flags the account with a "Needs Review" flag.

To create triggers for the Organization Sync Log, from Setup, enter *Org Sync Log* in the Quick Find box, then select **Triggers**.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions.

Example: The following trigger emails the administrator when a new entry has appeared in the log. Replace the text in **boldface** with your email address.

```
trigger NotifyOrgSyncEntry on PartnerNetWorkSyncLog (after insert) {
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(new String[] {'test@salesforce.com'});
    mail.setReplyTo('test@salesforce.com');
    mail.setSubject('New Org Sync Log Entry');
    mail.setPlainTextBody('There is a new entry in the Org Sync Log.');
    mail.setUseSignature(false);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
}
```

Reporting on the Organization Sync Log

As an Organization Sync administrator, you can create custom report types for the Organization Sync Log to stay informed about syncing failures.

When you create a custom report type, the Org Sync Log object can serve as the primary object or as the secondary object to any object supported in Organization Sync (for example, Accounts or Contacts). Here are some ways that you can use reporting to monitor the health of your Organization Sync environment:

- Create a custom report on the Org Sync Log object that shows all log entries from a particular date range.
- Create a custom report on the Case object that shows all log entries for records related to a case in the last month.
- Create a dashboard with reports that show:
  - The number of log entries in the past 24 hours
  - The number of log entries by field type

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: Because the log displays outgoing syncing events, the same report will show different results depending on which organization you're working in.

Example: Robert, a Salesforce administrator, wants to create a custom report type that shows all log entries for records related to an account in the last week:

- He creates a custom report type called Org Sync Log Account Entries and saves it in the Other Reports folder.

- He sets Account as the top-level object in the report type and Org Sync Log as the child object, using the **Each "A" record must have at least one related "B" record** relationship.
- Under Fields Available for Reports, he selects fields to add to the report type.
- He creates a report using his new custom report type, and selects **Last week** as the range.

Diagnosing Syncing Issues

Organization Sync users may encounter issues when working between the primary and secondary organization. To pinpoint the cause and resolve the problem, follow the troubleshooting steps.

EDITIONS

Available in: Salesforce Classic

Organization Sync is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

If a Synced Record is Missing from the Linked Organization...

Potential Reason	What to Do
The record is in the process of being copied to the linked organization.	In the organization where the record was created, check the Organization Sync Record Queue on the connection detail page. If the record is listed in the queue, it's being copied to the linked organization. The record disappears from the queue after it is successfully copied or the copying fails. If it fails, it's added to the Organization Sync Log. Consult the log for details.
The record's object type is not published and subscribed to.	1. Publish and subscribe to the object. 2. Perform a bulk sync to copy the records of that object type to the linked organization.
The record's parent record wasn't copied to the linked organization.	1. Confirm that the parent record's object type is published and subscribed to. 2. Manually forward the parent record to the linked organization via the External Sharing Related List on the record detail page. Once the parent record is forwarded, child-only records such as case comments are auto-forwarded, but other child records (such as contacts) must be manually forwarded.
The metadata for the record's object does not match between the primary and secondary organization. For example, a field or record type is missing from one organization.	Update the metadata of the organizations to match.
The record was not modified or created after the cutoff date that you selected when performing a bulk sync.	The bulk sync only copied records that were created or modified after the date that you select when performing a bulk sync. The earliest cutoff date that you can select is one year before

Potential Reason	What to Do
	the current date. If needed, you can manually forward the record to the linked organization via the External Sharing Related List.

If the Secondary Organization is Unavailable...

Potential Reason	What to Do
When the secondary organization undergoes maintenance, all record updates being sent from the primary organization are paused, and the Organization Sync Record Queue may accumulate a large number of items.	Wait for the maintenance to end. Once the secondary org is available again, the primary organization resumes sending record updates.

If a New User Isn't Being Copied to the Linked Organization...

Potential Reason	What to Do
A new profile was assigned to the user, and the profile hasn't been created in the linked organization.	The Organization Sync Log lists the user as a failed replication. To add the user to the linked organization, first add the profile assigned to the user: 1. Create the profile in the linked organization, making sure that it has the same name in both organizations. 2. In the organization where the user was created, navigate to the user record. In the user record, select **Edit** and click **Save** to trigger its replication.
The user doesn't have a standard Salesforce user license.	Only users with standard Salesforce user licenses are synced.

Suspend Your Organization Sync Connection

If you're experiencing repeated syncing failures or need to make significant changes in your organization, you can suspend the connection between your primary and secondary organizations. Updates made while the connection is suspended aren't synced until you resume the connection.

To suspend the connection between your organizations:

1. Navigate to your Organization Sync connection's detail page.
2. At the top of the page, click **Suspend**.
3. Click **OK**.
4. If you're using a two-way sync and want to suspend updates in both directions (recommended), complete the previous steps in the linked organization as well. Suspending a connection in only one direction may lead to syncing errors.

To resume the connection between your organizations:

1. Navigate to your Organization Sync connection's detail page.
2. At the top of the page, click **Resume**.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To suspend and resume the Organization Sync connection:

- "Manage Connections"

3. If you suspended updates in both directions, complete the previous steps in the linked organization as well.

When the connection is resumed, Salesforce sends updates in the queue to the linked organization. The time needed to process all the queued updates varies depending on how full the queue is.

Important: As the queue fills up, it'll take longer to get your two organizations back in sync. To minimize syncing issues, resume the connection as soon as possible.

Before You Suspend Your Organization Sync Connection

If needed, you can temporarily suspend the Organization Sync connection between your primary and secondary Salesforce organizations. Before suspending the connection, take a moment to learn how the process works.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

When should I suspend the connection?

Here are some reasons you may want to suspend the connection:

- You're experiencing a very high volume of sync failures, and need to diagnose and correct the issue.
- You need to make significant changes in one or both organizations that will prevent data from syncing successfully.

How does suspending the connection affect users?

For most users, nothing changes when the connection is suspended. Users with the "Manage Connections" permission will notice some changes on the detail page for the Organization Sync connection:

- A **Resume** button appears in place of the **Suspend** button.
- The Connection History log shows a recent `Connection Admin Suspended` event.
- The `Connection Owner` field and the options to sync metadata, users, and audit fields are updateable, but updates aren't sent to the linked organization until the connection is resumed.
- Depending on how long the connection has been suspended, the queue may contain more updates than usual.
- The Published Objects and Subscribed Objects related lists are hidden.

In addition, the Connection Owner and the user who suspended the connection receive an email twice per day that provides the number of events in the queue.

What happens when the connection is suspended?

All updates made while the connection is suspended are added to the Organization Sync Record Queue. Queued updates aren't sent to the linked organization until the connection is resumed.

Important: As the queue fills up, it'll take longer to get your two organizations back in sync. To minimize syncing issues, resume the connection as soon as possible.

Creating Apps

What Is a Salesforce App?

An *app* is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic apps and Lightning apps. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. You can customize both types of app to match the way your users work.

Classic apps are a collection of standard and custom tabs, including:

- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects
- Visualforce tabs
- Lightning component tabs
- Canvas apps via Visualforce tabs
- Web tabs

Lightning apps are a collection of items that include everything from the Classic apps list, plus Lightning Page tabs, and utilities like Lightning Voice. In Lightning apps, you can customize the app's logo and enhance its branding by customizing the color of the navigation bar.

You can also upgrade Classic apps to Lightning apps in Lightning Experience, but the two versions of the app must then be managed separately in their own environments.

Salesforce provides standard apps such as Sales and Call Center.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

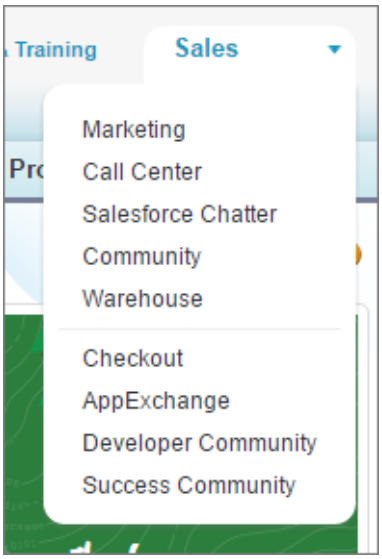
USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

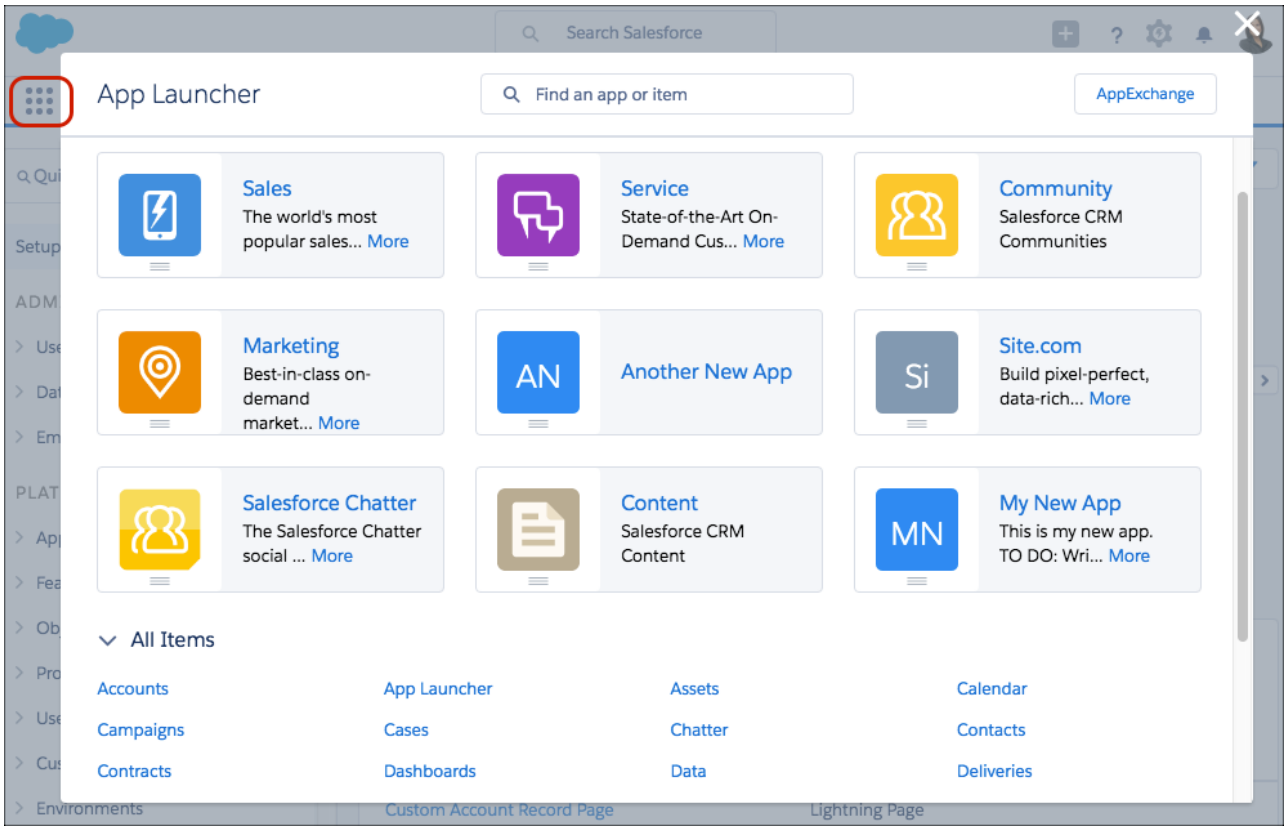
To manage apps:
- "Customize Application"

You can also build your own on-demand apps by grouping items into new custom apps. A custom app consists of a label, a description, and an ordered list of items, which often includes tabs. You can also add custom logos and branding to your custom apps.

In Salesforce Classic, custom apps are listed in the Force.com app menu, which is a drop-down list that displays at the top of every page.



In Lightning Experience, you can find your available custom apps in the App Launcher ( ).



When you choose an app, your screen changes to reflect the contents of that app. For example, if you switch from an app that contains Opportunities to another app that doesn't, the Opportunities item disappears. In addition, the app might display a different *default landing* tab when selected.

Apps are associated with profiles. Profiles control which tabs you can see or hide, as well as which apps are available to you.

SEE ALSO:

- Create Custom Apps for Salesforce Classic
- Set the Default Sort Order for Apps
- Salesforce App Considerations

Create Apps in Salesforce Classic with App Quick Start

App quick start is a fast way to create a basic Classic app in just one step.

1. From Setup, enter *Apps* in the Quick Find box, then select **Apps**, and click **Quick Start**. Alternatively, from the Force.com Home page, click **Add App** under **Getting Started**, or **App Quick Start** under **Quick Links**.
2. Enter the information needed for your app.

Field Name	Description
App Label	The app's name that appears in the Force.com app menu. The label can have a maximum of 40 characters, including spaces.
Plural Label	The plural name of the object. This name appears on the tab.
Singular Label	A name used to refer to the object in any user interface pages.
Gender	If it's appropriate for your org's default language, specify the gender of the label. This field appears if the org-wide default language expects gender.
Starts with a vowel sound	If it's appropriate for your org's default language, enable this option if your label should be preceded by "an" instead of "a".

3. Click **Create**.
4. On the **You're All Set!** page, click **here** to add new fields to your app.
5. To see your app as it will appear to users, click **Go To My App**.

The app quick start:

- Generates an app label and API name (a unique name that's used to refer to the object when using the Force.com API).
- Generates an object label and API name.
- Generates a tab label, and associates the tab with the object.
- Enables feed tracking for the object. Feed tracking lets people follow records of that object type and see Chatter feed updates.
- Enables access to the app and tab in your user profile. Any users who have the "Modify All Data" permission can also access the object.
- Generates a permission set that grants access to the new custom object.
- Assigns the permission set to the user who creates the app.

Note: If you're in a custom app, only the tabs included in the app appear and include the Create button.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create apps:

- "Customize Application"

  AND

  "Manage Profiles and Permission Sets"

After you've created an app, you can extend it with more components, specify access settings, and add users to your org.

SEE ALSO:

App Quick Start: Next Steps for Building and Managing Apps in Salesforce Classic

Salesforce App Considerations

App Quick Start: Next Steps for Building and Managing Apps in Salesforce Classic

After you've created a basic working app with app quick start in Salesforce Classic, build out the app with more objects and fields, define its access settings, and add users to share your app with them.

1. Build out your app with the basic components used in apps.
   - Create objects, which are custom database tables that allow you to store information specific to your app.
   - Create tabs that are associated with the objects you've created.
   - For each object, create fields to store the information that's important to your organization.
   - Create validation rules, which verify that the data users enter meets the standards you specify before they save a record.

   For quick shortcuts to these tools, use the Force.com quick access menu, which is available from object list view pages and record detail pages.
2. Create user profiles or permission sets. These are collections of settings and permissions that determine what users can do in an app.
3. Specify the types of access that users will have to the app.
   a. Make your app visible using profiles or permission sets.
   b. Make your object tabs visible.
   c. Set the object permissions for the objects you created.
4. Add users to your organization. When adding users, be sure to assign them the appropriate profiles or permission sets you created so they can access your app.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create objects, tabs, fields, and validation rules:
- "Customize Application"

To create users:
- "Manage Internal Users"

To create profiles and permission sets:
- "Manage Profiles and Permission Sets"

SEE ALSO:

Create Apps in Salesforce Classic with App Quick Start

Salesforce App Considerations

Create Custom Apps for Salesforce Classic

Create custom apps to give your Salesforce Classic users access to everything they need all in one place.

If you're new to custom apps, we recommend using Force.com quick start to create an app. With this tool, you can generate a basic working app in just one step.

If you've already created the objects, tabs, and fields you need for your app, follow these steps. With this option, you create an app label and logo, add items to the app, and assign the app to profiles.

1. From Setup, enter *Apps* in the `Quick Find` box, then select **Apps**.
2. Click **New**.
3. If the Salesforce console is available, select whether you want to define a custom app or a Salesforce console.
4. Give the app a name and description.

   An app name can have a maximum of 40 characters, including spaces.
5. Optionally, brand your app by giving it a custom logo.
6. Select which items to include in the app.
7. Optionally, set the default landing tab for your new app using the **Default Landing Tab** drop-down menu below the list of selected tabs. This determines the first tab a user sees when logging into this app.
8. Choose which profiles the app will be visible to.
9. Check the `Default` box to set the app as that profile's default app, meaning that new users with the profile see this app the first time they log in. Profiles with limits are excluded from this list.
10. Click **Save**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

SEE ALSO:

What Is a Salesforce App?

Salesforce App Considerations

Lightning Apps

With apps in Lightning Experience, members of your org can work more efficiently by easily switching between apps. Users can open apps you've created from the App Launcher. What's most important to sales reps? Accounts, events, and organizations. How about sales managers? Reports and dashboards make the top of the list. Lightning apps take things to another level past Classic apps by letting you brand your apps with a custom color, logo, and utility bar.

Lightning apps contain everything you'd expect from a custom app, such as custom and standard objects, and custom tabs. But Lightning apps can also include Lightning Page tabs and utilities like Lightning Voice. The navigation model of Lightning apps is optimized for efficiency, with actions on certain items like Opportunities in the navigation bar.

Custom apps from Salesforce Classic automatically work in Lightning Experience and can be upgraded to Lightning apps. Classic apps appear in the list of apps in Setup alongside your Lightning apps, and are available from the App Launcher as long as the `Show in Lightning Experience` attribute is enabled.

However, the reverse is not true for Lightning apps. Lightning apps aren't available in Salesforce Classic.

EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

You can assign multiple user profiles to multiple apps. Also, you can assign as many user profiles to one app as you need to. For instance, you have several groups involved with inside sales. Assign all the groups to your inside sales app, and they all have access to it.

To switch between apps, users can use the App Launcher. This makes it easy for users to switch contexts and still have access to the items, objects, and pages they need most.

Note: In Salesforce Classic, users can customize the tabs that display in each app they have access to. Customize My Tabs functionality isn't supported in Lightning Experience. Any customized tab settings a user has made to a Classic app aren't reflected in Lightning Experience.

You can view all the apps in your org from the App Manager. In Lightning Experience Setup, enter *App* in the `Quick Find` box, then select **App Manager**.

SEE ALSO:

Create Lightning Apps

Upgrade Classic Apps to Lightning Apps

Salesforce App Considerations

Tips for Creating Apps in Lightning Experience

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Professional**, **Performance**, **Unlimited**, and **Developer** Editions

It's time for the fun part: deciding how to set up Lightning apps for your users. Here are some tips for planning Lightning apps for your org.

The best time to create Lightning apps is when you're rolling out Lightning Experience. So make creating Lightning apps a part of your rollout strategy. Check out the Trailhead module "Lightning Experience Rollout" for many great ideas to help you make a smooth transition.

Talk to your users. Ask them what their priorities are. Customizing tabs in apps gives you a unique opportunity to engage with your users. Each group of users has its own priorities. Find out which objects and items represent their highest priorities.

- Ask users to post feedback to a Chatter group.
- Publish polls.
- Schedule lunch sessions. Everyone likes a free lunch, and nearly everybody is happy to express their opinion.

Create a master list of objects that everyone in your org wants. Then trim down the list for each group—sales reps, sales managers, execs, and so on. The menus for every user group share some common objects, like Home, Tasks, and Feed. Keep the high-priority items for each group at the top. Put low-priority items at the bottom, or remove them altogether. Users can always go to the App Launcher to get the items they use less often.

Create Lightning Apps

As in Salesforce Classic, you can create apps in Lightning Experience, but with even more bells and whistles. You can brand and customize Lightning apps to help your users work more efficiently. For example, you can create a Lightning app for your finance department that includes all the important items (including tabs) users need to complete common tasks. You can customize the navigation bar color, brand it with a logo, and make the app available in the App Launcher for the user profiles associated with the finance department.

1. From Setup, enter *App* in the `Quick Find` box, then select **App Manager**
2. Click **New Lightning App**, and walk through the New Lightning App wizard.

   You can give your app a name, set its primary color, give it a logo, customize which items appear in the app's navigation bar, and assign the app to user profiles. If your org includes utilities like Lightning Voice or Open CTI Softphone, you'll see options to add them to your app's utility bar.

   When organizing the navigation bar, keep in mind that the item at the top of the list becomes your app's landing page.

   Tip: The app description displays alongside the icon in the App Launcher. Make the description meaningful to your users.

EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

SEE ALSO:

Utility Bar Implementation Guide

Lightning Apps

Salesforce App Considerations

Lightning App Navigation Bar Items

Most of the items that appear in the App Launcher can appear in a Lightning app navigation bar. To add items to an app's navigation bar, you can use the Lightning app creation wizard, which lets you choose from a list of available items.

The list of available items contains only those items in your org that are eligible for Lightning app navigation bars, which includes:

- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects and apps
- Visualforce tabs
- Lightning component tabs
- Lightning page tabs
- Canvas apps via Visualforcetabs
- Web tabs
- Utilities like Lightning Voice

Note: You can't add Connected apps like Gmail™ and Microsoft® Office 365™ to the navigation bar. Users can continue to access them from the App Launcher.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Professional**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view navigation menus:
- "View Setup and Configuration"

To create navigation menus:
- "Customize Application"

Upgrade Classic Apps to Lightning Apps

You can upgrade a Classic app to a Lightning app in Lightning Experience, enhancing it for your Lightning Experience users with a customized color, logo, utility bar, and additional items like Lightning Pages supported in the navigation bar.

1. From Setup, enter *App* in the `Quick Find` box, then select **App Manager**.
2. Find the Classic app that you want to upgrade in the apps list.

   Note: A checkmark in the Visible in Lightning column means that the app is accessible in Lightning Experience via the App Launcher and is fully functional. Even though a Classic app works in Lightning Experience, it doesn't take advantage of all the benefits of being a Lightning app. That's why we recommend that you upgrade it.

3. Click , and select **Upgrade**.
   Your Classic app is copied and upgraded for Lightning Experience. You now have two versions of the app: a Classic version, and a Lightning version. After you upgrade it, the Classic app is no longer accessible in Lightning Experience via the App Launcher. You'll still see the Classic app in the apps list, but with the Visible in Lightning column deselected.

The two versions of your app now must be managed separately. Future changes you make to the Classic app won't be reflected in the Lightning version of the app, and vice versa.

Note: You can toggle the availability of your Classic apps in Lightning Experience by selecting or deselecting `Show in Lightning Experience` on the Classic app's detail page.

EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

SEE ALSO:

Lightning Apps

Salesforce App Considerations

Salesforce App Considerations

Keep these considerations in mind when working with apps in either Lightning Experience or Salesforce Classic.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

General

- You can delete custom apps, but not standard apps. Deleting a custom app removes it from the Force.com apps menu and the App Launcher, but doesn't delete any associated objects. If you've created objects for an app, consider deleting them as well.
- Salesforce console apps are custom apps.
- You can't change the label, description, or logo for a standard app.
- You can't change an app's type—such as standard to connected or vice versa—after you've created it.
- For Salesforce Platform and Salesforce Platform One license users, the Platform standard app is the only app listed in the Force.com app menu and the Lightning Experience App Launcher. For details about specifying a unique label for the Platform standard app in Salesforce Classic, see Create Custom Apps for Salesforce Classic on page 1042.
- To assign apps to user profiles in Professional Edition, you must have user profiles enabled for your org.

Classic Apps

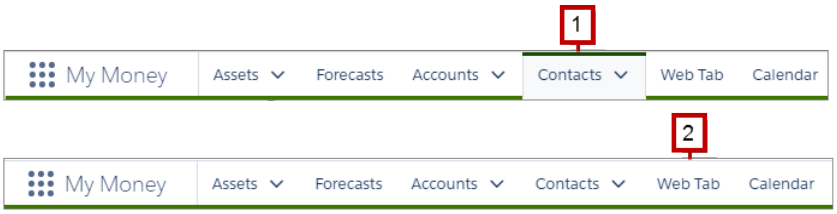
- Consider these requirements when choosing a custom app logo for a Classic app from the document library:
  - The image must be in GIF or JPEG format and less than 20 KB.
  - If the image is larger than 300 pixels wide by 55 pixels high, then it is scaled to fit.
  - For the best on-screen display, we recommend that you use an image with a transparent background.
  - The `Externally Available` checkbox must be selected on the document's properties so that users can view the image.

Lightning Apps

- The number of Lightning Apps you can create in an org varies by edition.

Edition	Lightning Apps Limit
Professional Edition	10
Enterprise Edition	25
Unlimited Edition	Unlimited

- In Salesforce Classic, users can customize the tabs that display in each app they have access to. Customize My Tabs functionality isn't supported in Lightning Experience. Any customized tab settings a user has made to a Classic app aren't reflected in Lightning Experience.
- If you enable Lightning Voice or other default utility bar items in an app, you can't disable them. You also can't go back and enable them after you've created the app.
- A Lightning app's description displays in the App Launcher, so we recommend that you keep the description concise.
- Consider these requirements when choosing a custom app image for apps in Lightning Experience:
  - App images represent your app in both Lightning Experience and Salesforce1.
  - Choose a JPG, PNG, BMP, or GIF image that's smaller than 5 MB.
  - For best results, upload an image that's 128 by 128 pixels. Images larger than the maximum display of 128 by 128 pixels are automatically resized.
- Not all objects that appear in the App Launcher can appear in an app, but it's easy to figure out which ones can. When you start the wizard from the Lightning Experience App Manager, you see all available items for a navigation bar.
- You can create records and access recent records and lists for certain items directly from the navigation bar. Items with next to their name support this feature, with a few exceptions. Tasks and Notes allow you to create a record but you can't access recent records or lists. Reports and Dashboards allow you to see recent records but you can't see recent lists or create a record.
- Some tabs, such as web tabs and Visualforce tabs, aren't highlighted when you select them on the navigation bar. For example, when you select **Contacts**, the tab is highlighted (**1**). However, when you select a web tab, the page displays but the tab isn't highlighted (**2**).



What is a Subtab App?

An *app* is a group of tabs that work as a unit to provide application functionality. Similarly, a *subtab app* is a collection of tabs that appears on the Chatter profile page. A subtab app can include both default and custom tabs.

Users can see different sets of tabs on the profile page depending on their context. Subtab apps are the various sets of tabs available on specific pages, such as users' profile pages.

These default subtab apps determine which tabs display, depending on the user's context.

Subtab App	Displayed to the user when viewing...
Profile (Others)	Another user inside their internal organization
Profile (Self)	Their own profile inside their internal organization
Profile in Communities (Others)	Another user while inside a community. It's shown only if Communities is enabled.
Profile in Communities (Self)	Their own profile inside a community. It's shown only if Communities is enabled.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: End users can't customize the display of subtab apps. Administrators can hide tabs within subtab apps using the `Tab Hidden` option in Tab Settings. Users can see tabs set to `Default Off` and `Default On`.

SEE ALSO:

- Managing Subtab Apps
- Controlling Subtab App Visibility
- Tip sheet: Customizing Chatter Profile Pages

Managing Subtab Apps

You can view and customize the subtab apps on users' profile pages.

From Setup, enter *Apps* in the Quick Find box, then select **Apps** to display your organization's subtab apps.

You can do the following:

- To view details for a subtab app, click the name in the **Subtab Apps** section. This displays its properties, such as which tabs are part of the app, including any tabs that are not yet deployed. Click custom tabs in the Included Tabs list to view details.
- To change the properties of a subtab app, click **Edit** to choose the tabs to include in the subtab app, change their display order, and set the Default Landing Tab.

  Note: Administrators can change permission sets or profile settings to limit users' access to each tab. This allows administrators to make specific tabs available to some users, but not to others.

SEE ALSO:

What is a Subtab App?

Controlling Subtab App Visibility

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

Controlling Subtab App Visibility

Once you have configured subtab apps, you can specify which users can see specific tabs on the profile page.

To control the visibility of tabs within a subtab app:

1. From Setup, enter *Profiles* in the Quick Find box, then select **Profiles**.
2. Do one of the following:
   - Original profile user interface—Click **Edit** next to the profile you want to modify and scroll to the Tab Settings section.
   - Enhanced profile user interface—Click the profile you want to modify and click **Object Settings**. Click the object you want to modify and click **Edit**.

   Note: Some profiles, including Chatter External and Chatter Free users, don't have the permissions needed to view subtab apps.
3. Change the tab settings.

   End users can't customize the display of subtab apps. Administrators can hide tabs within subtab apps using the Tab Hidden option in Tab Settings. Users can see tabs set to Default Off and Default On.
4. (Original profile user interface only) To reset users' tab customizations to the tab visibility settings that you specify, select **Overwrite users' personal tab customizations**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

5. Click **Save**.

SEE ALSO:

What is a Subtab App?

Managing Subtab Apps

Set the Default Sort Order for Apps

As a Salesforce admin, you control the default sort order of the Salesforce standard, custom, and connected apps that your users see in your org. Users can then rearrange their apps on the App Launcher to get their liking. You can also hide apps so that they don't show in the App Launcher.

Apps include Salesforce standard apps, such as the Salesforce Marketing app, the Call Center app, and any custom apps that you created for your org. Connected apps are third-party apps, such as Gmail, Google Drive, and Microsoft Office 365, that you install to make it easy for your users to get their work done.

EDITIONS

Available in: both Lightning Experience and Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

IN THIS SECTION:

Reorder the App Menu and App Launcher in Salesforce Classic

You can change the order in which apps appear in the app menu and App Launcher. The app menu is a drop-down list in the upper-right corner of every page in Salesforce Classic. If enabled, the App Launcher is listed in the drop-down menu. Apps in the App Launcher appear as large tiles and link to Salesforce standard apps, custom apps, and connected apps.

Reorder the App Launcher Apps in Lightning Experience

As a Salesforce admin, you can change your org's default visibility and the order in which apps appear in the Lightning Experience App Launcher. Users can then reorder their personal view of the App Launcher to their liking.

Reorder the App Menu and App Launcher in Salesforce Classic

You can change the order in which apps appear in the app menu and App Launcher. The app menu is a drop-down list in the upper-right corner of every page in Salesforce Classic. If enabled, the App Launcher is listed in the drop-down menu. Apps in the App Launcher appear as large tiles and link to Salesforce standard apps, custom apps, and connected apps.

1. From Setup, enter *App Menu* in the `Quick Find` box, then select **App Menu**.
2. From the list of app menu items, drag the apps to change their order. Changes take effect immediately.
3. Optionally, click **Visible in App Launcher** or **Hidden in App Launcher** to show or hide individual apps from the App Launcher for all users in the org.

The app menu lists all apps installed in the org. However, the apps that users see in their App Launcher vary. Salesforce admins control each app's visibility settings and users' permissions.

EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:

- "View Setup and Configuration"

To manage apps:

- "Customize Application"

SEE ALSO:

What Is a Salesforce App?

Create Custom Apps for Salesforce Classic

Reorder the App Launcher Apps in Lightning Experience

As a Salesforce admin, you can change your org's default visibility and the order in which apps appear in the Lightning Experience App Launcher. Users can then reorder their personal view of the App Launcher to their liking.

The App Launcher displays a user's available Salesforce apps and the connected apps that a Salesforce admin installs for the org.

1. From Setup, enter *App Menu* in the Quick Find box, then select **App Menu**.
2. From the list of app menu items, drag the apps to change their order. Changes take effect immediately.
3. Optionally, click **Visible in App Launcher** or **Hidden in App Launcher** to show or hide individual apps from the App Launcher for all users in the org.

All apps installed in the org appear on the app menu items list. However, the apps that users see in their app menu and App Launcher vary depending on each app's visibility settings and the user's permissions. Users see only the apps that they are authorized to see according to their profile or permission sets.

For connected apps and service providers to appear in the App Launcher, specify their start URL in the App Manager.

EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view apps:
- "View Setup and Configuration"

To manage apps:
- "Customize Application"

SEE ALSO:

What Is a Salesforce App?

Make the App Launcher the Default Landing Page

Make it easy for your Salesforce Identity users to access what they need by presenting the redesigned App Launcher as the default landing page when they log in to Salesforce.

Note: These steps work in Lightning Experience. If you see the App Launcher icon ( ) on the left side of the navigation bar at the top of your screen, you're in Lightning Experience. If not, you're in Salesforce Classic.

1. From Setup, enter *App Manager* in the Quick Find box, then select **App Manager**.
2. Click **New Lightning App** and walk through the New Lightning App wizard.

   Add only the App Launcher tab to Selected Items.
3. Make the App Launcher the default when users log in for the first time.
   a. From Setup, enter *Profiles* in the Quick Find box, then select **Profiles**.
   b. Select a profile and scroll to the Custom App Settings section.
   c. Select **Default** next to the Lightning app.
4. Log out and log in again.

   The new Lightning app appears in the navigation bar and App Launcher.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Force.com Home Page

The Force.com Home page contains options for building and managing applications.

You access the Force.com Home page from Setup.

- The left sidebar, which you can browse or search, provides access to all setup actions, tasks, and tools.
- The Getting Started box contains a tool for generating a basic app in a single step and links to information about extending and managing apps. This box doesn't appear if you've previously dismissed it.
- The Recent Items list shows recent metadata items that you've viewed, edited, or created and their related objects.
- The System Overview messages box displays messages to remind you when your organization reaches its usage limits. The System Overview messages box is not enabled by default.
- The Quick Links box provides links for managing tools, users, apps, security, and data.
- The Community box showcases available resources. If you've previously dismissed this box, it reappears with each new release.
- The right pane includes external links that are useful for developers and administrators.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions except **Database.com**

USER PERMISSIONS

To access the Force.com Home page:
- "Customize Application"

SEE ALSO:

Recent Items List (Beta)

Recent Items List (Beta)

The Recent Items list shows recent metadata items that you've viewed, edited, or created and their related objects.

Note: The Recent Items list is in beta. It is production quality but has known limitations.

The Recent Items list includes:

- Apex classes
- Apex triggers
- Approval processes
- Apps
- Custom report types
- Email templates
- Fields
- Objects
- Page layouts
- Permission sets
- Profiles
- Record types
- Static resources
- Tabs
- Users

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions except **Database.com**

- Validation rules
- Visualforce pages
- Visualforce components
- Workflow email alerts
- Workflow field updates
- Workflow outbound messages
- Workflow rules
- Workflow tasks

Note: The Recent Items list in Setup is independent of the Recent Items section in the sidebar column of many Salesforce pages. The list in Setup shows items that administrators use, while the Recent Items section in the sidebar displays records with which end users have worked.

Configuring Force.com System Overview Messages

Note: The system overview page shows only the items enabled for your organization. For example, your system overview page shows workflow rules only if workflow is enabled for your organization.

Add system overview usage messages to the Force.com Home page to remind you when your organization approaches its limits. You can expand, collapse, and dismiss the system overview messages that appear on the Force.com Home page. By default, the system overview home page messages are enabled.

To configure the system overview messages on the Force.com Home page:

1. From Setup, enter *System Overview* in the Quick Find box, then select **System Overview**.
2. Click **Configure Force.com Messages**.
3. Select or deselect the types of system overview messages to show or hide on the Force.com Home page.
4. Click **OK**.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except **Personal** and Database.com

USER PERMISSIONS

To configure Force.com messages:
- "Customize Application"

Important: System overview messages only appear on the Force.com Home page when your organization approaches its limits.

When you enable or dismiss system overview messages on the Force.com Home page, it only impacts your individual view of the system overview messages.

Create Custom Mobile Apps

Create Custom Mobile Apps with Lightning Pages and the Lightning App Builder

Lightning App Builder

The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for Salesforce1 and Lightning Experience. With the Lightning App Builder, you can combine various components on a single page to give your users what they need all in one place.

You can access the Lightning App Builder from Setup by entering *Lightning App Builder* in the Quick Find box and then selecting **Lightning App Builder**.

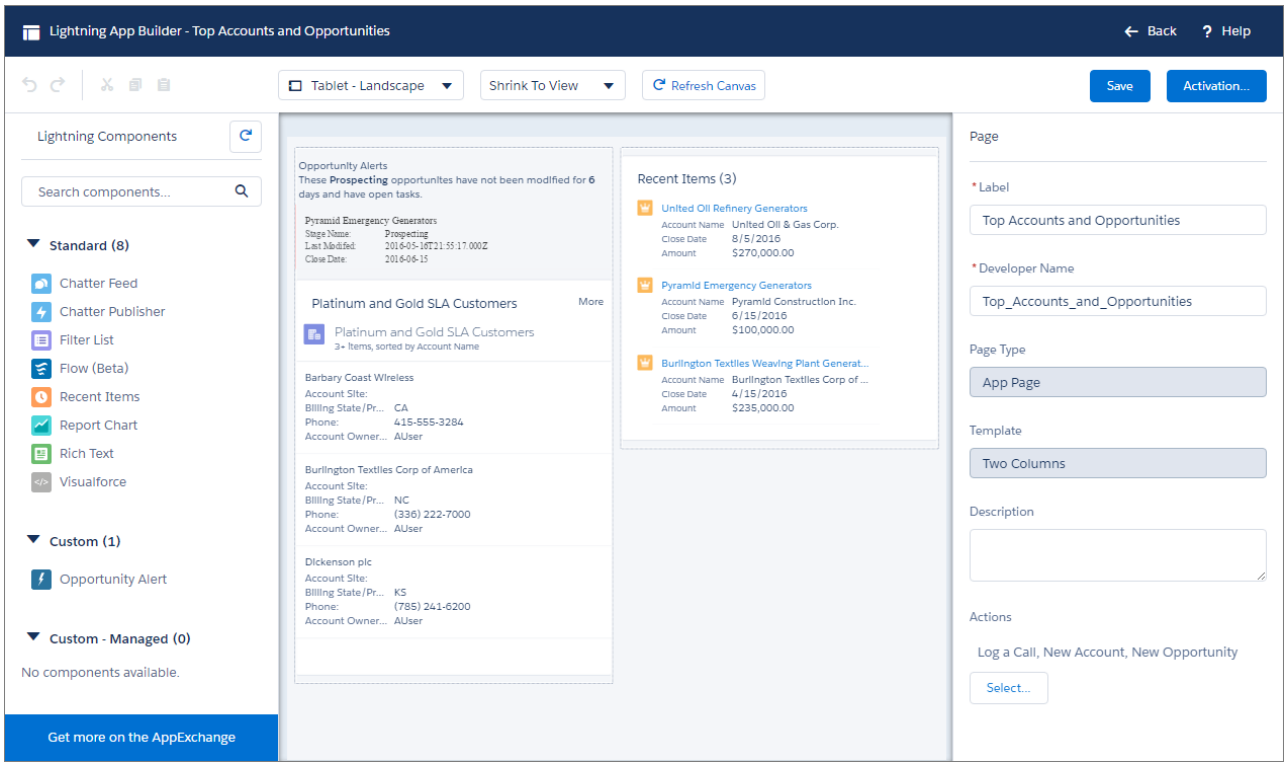
With the Lightning App Builder, you can build:

- Single-page apps that drill down into standard pages
- Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
- "Point" apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they've submitted
- Custom record pages for your objects, tailored to the needs of your users
- Custom Home pages containing the components and features that your users use most

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



Lightning Pages are the underlying technology for the Lightning App Builder. A Lightning Page is a custom layout that lets you design pages for use in the Salesforce1 mobile app or in Lightning Experience.

Lightning Pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

You can use standard, custom, and third-party components in the Lightning App Builder.

Standard Components
Standard components are Lightning components built by Salesforce.

Custom Components
Custom components are Lightning components that you or someone else have created. With some modifications, custom Lightning components can work in the Lightning App Builder. For more information on making Lightning components App Builder–friendly, see the *Lightning Components Developer Guide*.

Third-Party Components on AppExchange
The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.

The Lightning App Builder is supported on the latest desktop versions of Microsoft® Internet Explorer®, Google Chrome™, Apple® Safari®, and Mozilla® Firefox®. Lightning App Builder is not supported on mobile browsers or on Internet Explorer versions 10 and earlier. The minimum recommended resolution for Lightning App Builder is 1280x1024.

Create a Custom App Page: The Big Picture

With just a few steps, you can create an app page that lets your Lightning Experience and Salesforce1 users access the most important objects and items in your custom app. Custom app pages are built using Lightning Pages.

1. Before creating your page, determine which components you want to include and the global actions your users need.
2. Create the Lightning Page.
3. Add actions to your page.
4. Activate your Lightning Page in the Lightning App Builder.

   Activation lets you create a custom tab for your app page, set its visibility, and add it to the Salesforce1 navigation list and Lightning Experience app navigation bars all in one place.

Lightning Pages

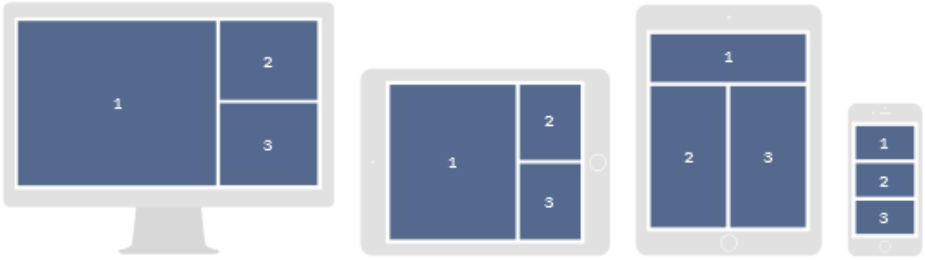
A Lightning Page is a custom layout that lets you design pages for use in the Salesforce1 mobile app or in Lightning Experience.

Lightning Pages occupy a middle ground between page layouts and Visualforce pages. Like a page layout, Lightning Pages allow you to add custom items to a page. However, these items, instead of being fields or Visualforce components, are Lightning components, which allow much more flexibility.

You can use a Lightning Page to create an app page that you can add to the Salesforce1 navigation list or to a Lightning app's navigation bar. An app page enables your users to quickly access the objects and items that are most important in that app.

You can also use a Lightning Page to create a customized record or Home page for Lightning Experience.

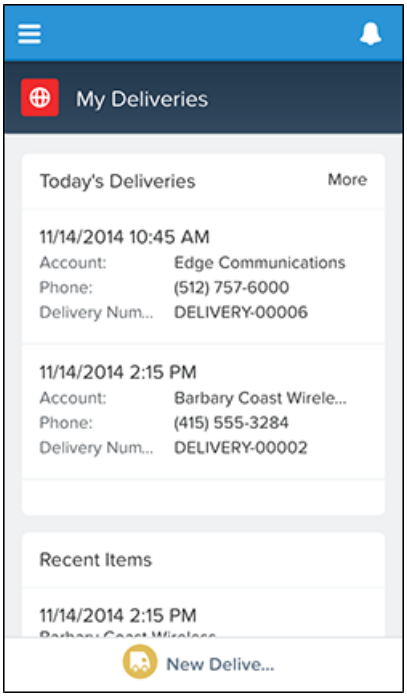
The structure of a Lightning Page adapts for the device it's viewed on. The template you choose when creating the page controls how it displays on a given device. The Lightning Page's template divides the page into regions.



Lightning Pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

A Lightning Page consists of regions that can contain multiple components.

Example: This Lightning Page in Salesforce1 has a filter list component, a recent items component, and one global action.



SEE ALSO:

Considerations and Limitations for Flows in Lightning Pages (Beta)

Lightning Page Types

You can create different types of Lightning Pages with the Lightning App Builder. After you set a Lightning Page's type, you can't change it.

App Page

App pages are supported both in Salesforce1 and Lightning Experience.

Use an App Page to create a home page for a third-party app that you can add directly into the Salesforce1 and Lightning Experience navigation menus. Your users then have an app home page that lets them quickly access the objects and items that are most important in that app.

You can add global actions to an app page to enhance its functionality. Global actions allow a user to do things from your app page, such as add call details, create and update records, send email, and create a task. When a user visits a Lightning Page in Salesforce1, the page's actions appear in its action bar. In Lightning Experience, actions appear in the highlights panel at the top of the page.

Important: Create actions in the full Salesforce site before adding them to your app page.

App pages support only global actions. Standard Chatter actions, such as Post, File, Link, and Poll, aren't supported.

When a user visits an app page in Salesforce1 or Lightning Experience, only the actions that you specify for the page are displayed. If you haven't specified any actions, no actions appear.

Home Page

Create Home pages with features relevant to specific types of users, and assign the customized pages to different user profiles. Custom Home pages are supported in Lightning Experience only.

Record Page

With a Record Page, you can create a customized version of an object's record page, tailoring it to your users' needs. Custom record pages are supported in Lightning Experience only.

Note: Actions you see on Lightning Experience record pages and the Home page are taken from object and global page layouts. You can't add, edit, or remove actions on these pages using the Lightning App Builder.

Email Application Pane (Beta)

Create custom Email Application Panes that let users work with Salesforce content that's most relevant to them from Lightning for Outlook. Custom Email Application panes are supported in Salesforce Classic and Lightning Experience.

Standard Lightning Page Components

Standard components are Lightning components built by Salesforce. Several standard components are available when creating Lightning Pages.

Note: The components listed here are supported across all page types, except when otherwise indicated. This is not a comprehensive list of all available standard components. Other standard component types are available and vary based on the type of page you're creating and which object the page is associated with.

Some standard components have required properties that you must configure before the component will work on the page. When you add a component to a Lightning Page in the Lightning App Builder, its required properties are marked with an asterisk.

A Lightning Page region or tab can contain up to 25 components.

If you have an app page that is intended for use in Salesforce1, you can associate Lightning Page components on that page only with objects that Salesforce1 supports. For a list of supported objects, see What Data Is Available in the Salesforce1 Mobile App?.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

App Launcher Component

The App Launcher displays a user's available Salesforce apps and any connected apps the administrator configures. Users navigate between apps using the App Launcher.

This component is supported in API version 35.0 and later.

Contact Salesforce to enable the App Launcher component for the Lightning App Builder in your organization.

Chatter Component

As of API version 38, the Feed component has been renamed Chatter in the Lightning App Builder. Use it to place a publisher and feed combo on a record page.

Chatter Feed Component

Use the Chatter Feed component to place a feed anywhere on a record page. The feed gives you a way to view posts, comments, and questions. Its one attribute, `Feed Type`, takes one of these values:

- `Bookmarked` shows a feed of all items the current user has bookmarked.
- `What I Follow` shows a feed of all items the current user has followed.
- `To Me` show a feed of all items where the current user is mentioned.

No special coding is required to join a feed to a publisher; the connection is made automatically with the publisher and any feed on the page. The Chatter Feed component is available in API version 38.0 and later.

Chatter Publisher Component

Use the Chatter Publisher component to place a feed publisher anywhere on a record page. The publisher gives you a way to post, poll, or ask a question in a feed. Its one attribute, `Type`, always takes the value *`Global`*. Use the Chatter Publisher component with the Chatter Feed component to get a full feed experience. No special coding is required to join a publisher to a feed; the connection is made automatically with the publisher and any feed on the page. The Chatter Publisher component is available in API version 38.0 and later.

Filter List Component

The Filter List component points to a list view and displays the first few records from that view. It supports all public and shared list views that are associated with standard and custom objects, except:

- Activity
- ContentVersion (Files)
- Task
- User
- UserProfile

By default, a Filter List component displays the first three records in the list, but you can set it to show a maximum of 30.

You can't give a Filter List component a custom name. The component's name is derived from the name of the list view filter you select when you configure the component.

In the Lightning App Builder, the Object drop-down list for this component displays only those objects that have list views associated with them.

Highlights Panel Component

The Highlights Panel component displays record highlights along with action buttons. To streamline your highlights panel by not showing the second row of information, click **Collapsed**. To show fewer action buttons, reduce the number using the **Number of Visible Actions** attribute.

To display the highlights horizontally or vertically, drag the Highlights Panel component into a region with the horizontal or vertical dimensions you want. The highlights panel adjusts to fit the region's space. For example, if you drag it into a narrow column, the highlights display vertically. If you drag it to a full-page width column, the highlights display horizontally.

Recent Items Component

The Recent Items component displays a list of the most recently used items. The default is three, but you can set it to show a maximum of 30. In the Lightning App Builder, you can specify which objects' records appear in the recent items list.

The Recent Items component supports these objects, based on the specified properties:

- All custom objects
- All standard objects for which both of these conditions are true:
  - A compact layout is defined for the object.
  - The object is tracked in the most recently used objects list.

  If an object is tracked in the most recently used objects list, one or both of the `LastViewedDate` or `LastReferencedDate` fields are present.

The Recent Items component doesn't support these objects:

- ContentVersion (Files)
- User
- UserProfile

Report Chart Component

Use the Report Chart component to include a chart from a report in your Lightning Page. If you leave the component's `Label` field blank, the component's label is derived from the report's label.

The chart refreshes if its report data is more than one day old. In the component properties, you can choose to display a refresh button to enable users to refresh the chart. Saving the report's definitions also updates the chart data in the component.

Setting a filter on the report chart data is supported only for record pages. If you set a filter option, the Report Chart component displays only that filtered data to users.

This component is supported in API version 32.0 and later. It doesn't work with reports that are located in the My Personal Custom Reports folder. Report Chart components that refer to reports in the Unfiled Public Reports folder aren't deployable if you include them in a package.

Rich Text Component

Use the Rich Text component to add text and simple HTML markup to your Lightning Page.

Note: JavaScript, CSS, iframes, and other advanced markup aren't supported. To use advanced HTML elements in a component, we recommend using a Visualforce page component or a custom Lightning component.

The Rich Text component is limited to 4,000 characters. This component is supported in API version 32.0 and later.

Tabs Component

Use the Tabs component to add a set of tabs to a region of your Lightning Page. Choose from a set of standard tabs or create custom tabs to enhance record pages for your Lightning Experience users. The Tabs component is supported only for Lightning Experience record pages.

You can place up to 25 tabs in a Tabs component. This component is supported in API version 36.0 and later.

Visualforce Page Component

Use the Visualforce Page component to include a Visualforce page in your Lightning Page.

If you leave the component's `Label` field blank, the component's label is taken from the label of the Visualforce page that you assign to it.

If you leave the `Height` field blank, the Visualforce page's height defaults to 300 pixels when it displays in Salesforce1.

This component is supported in API version 32.0 and later.

To appear in Salesforce1, the Visualforce page must have the `Available for Salesforce mobile apps and Lightning Pages` option selected. This option is available for pages that are set to API version 27.0 and later.

Wave Analytics Dashboard Component

Use the Wave Dashboard component to include Wave Analytics dashboards in your Lightning Page. Contact Salesforce to enable the Wave Dashboard component for the Lightning App Builder.

Select the dashboard to display from the drop-down list. If you leave the `Height` field blank, the dashboard's height defaults to 300 pixels when it displays in your Lightning Page.

You can control the visibility of the dashboard's title and specify whether the dashboard appears if an error occurs. With the Open Links in New Windows attribute, you can specify where links from the dashboard to other assets are opened. With the Filter attribute, you can use JSON to filter dataset fields at runtime.

For more information, see Embed Wave Dashboards in Lightning Pages.

Custom Lightning Page Components

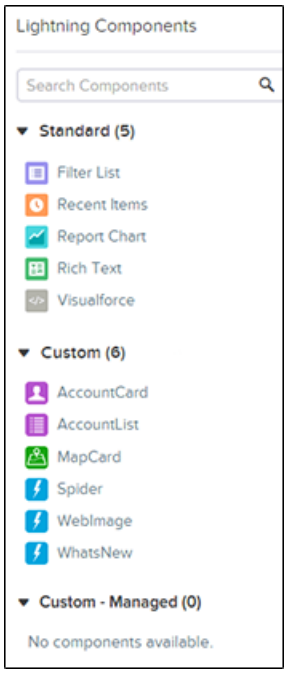
The Lightning App Builder supports custom Lightning components.

Custom components in your org that are configured for use in the Lightning App Builder appear in the Lightning Components pane.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



If you have an app page that is intended for use in Salesforce1, you can associate Lightning Page components on that page only with objects that Salesforce1 supports. For a list of supported objects, see What Data Is Available in the Salesforce1 Mobile App?.

Your custom Lightning components don't automatically work onLightning Pages or in the Lightning App Builder. To make a custom component usable in both, you need to:

1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning Pages.
2. Deploy My Domain in your org. When you deploy My Domain, references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

Create an App Home Page with the Lightning App Builder

Use an App Page type of Lightning Page to create a home page for an app that you can then add directly into the Salesforce1 navigation list and into Lightning apps. Your users can then easily access the objects and items that are most important in that app.

1. From Setup, enter *App Builder* in the Quick Find box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **App Page** and then click **Next**.
4. Select a page template, and then click **Next**.
5. Give your app page a label.

   The label can be up to 80 characters.
6. Click **Finish**.
7. Drag the components that you want onto the page.

   Drag components up and down to rearrange them. You can also click a component's top or bottom border to create an insertion point ( ) for the next component.
8. Click each component on the page to configure its properties.
9. Click in the empty area of the canvas or on the **Page** link in the breadcrumb to configure the page properties.

   The Description field is limited to 255 characters. The Developer Name field is limited to 80 characters, but if you have a namespaced org, we recommend using fewer than 65 characters. When you create a Lightning Page, the developer name is derived from the first 40 characters of the label that you give the page.
10. Optionally add global actions to your page.
    a. In the page properties, click **Select**.
    b. Add, remove, or reorder the actions for your page.

       Actions you select appear on the page's action bar in Salesforce1 and in the highlights panel at the top of the page in Lightning Experience. An app page is the only type of Lightning Page that you can add actions to in this way. Other Lightning Pages derive their actions from the object and global page layouts.

    c. Click **OK**.
11. Click **Save** when you're done editing your page.

To give your users access to the app page in Salesforce1 and Lightning Experience, you must activate it first.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder
- "Customize Application"

To view Lightning Pages in the Lightning App Builder
- "View Setup and Configuration"

Add Your App Page to Salesforce1 and Lightning Experience

To add your custom app page to Salesforce1 and Lightning Experience, you must activate it. You can also rename the Lightning Page tab, adjust its visibility, set its position in the Salesforce1 navigation list, and add it to a Lightning app. The Lightning App Builder's activation feature makes this process easy.

Tip: You can activate your app page for either Salesforce1, Lightning Experience, or both.

1. To open your app page, from Setup, enter *Lightning App Builder* in the Quick Find box, select **Lightning App Builder**, and then click **Edit** next to the page.
2. In the Lightning App Builder, click **Activation**.
3. Update the activation properties, if desired.

   You can:

   - Change the page's custom tab label. By default, the label that you give the Lightning Page is used as the label for its custom tab.
   - Change the custom tab's icon. The icon that you choose here is used as the icon for the app in Salesforce1 and for the page in Lightning Experience.
   - Adjust the custom tab's visibility if you're activating the page for the first time.
4. Add the page to one or more custom Lightning apps.

   You must create a custom Lightning app before you can add your app page to Lightning Experience.
5. Set the page's position in the Salesforce1 navigation list.

   To have your page appear in the Apps section of the Salesforce1 navigation list, put the tab below Smart Search Items.

   Note: The first item that you put in the navigation list becomes your users' landing page. If you want your users to see your custom app page first when they open Salesforce1, put the page first in the list.
6. Click **Activate**.

Your Lightning Page is now ready for your Salesforce1 and Lightning Experience users!

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder
- "Customize Application"

To view Lightning Pages in the Lightning App Builder
- "View Setup and Configuration"

Create Lightning Page Tabs

Before you can include an App Page type of Lightning Page in the Salesforce1 navigation list or in a Lightning app, you must create a custom tab for it.

When you first activate an app page in the Lightning App Builder, a tab is created for the page as part of the activation process. You can also create a tab for the page manually in Setup before you activate it.

Lightning Page tabs only facilitate inclusion of the Lightning Page in Salesforce1 and Lightning Experience.

You can create a custom tab only for an App Page type of Lightning Page.

1. From Setup, enter *Tabs* in the Quick Find box, then select **Tabs**.
2. Click **New** in the Lightning Page Tabs related list.
3. Choose a Lightning Page for the tab.
4. Enter a label.

EDITIONS

Available in: Salesforce Classic

Available in: **All** editions except **Database.com**

USER PERMISSIONS

To create and edit custom tabs:
- "Customize Application"

This text is used as the display name for the Lightning Page.

5. Select a tab style to set a color scheme and icon for the Lightning Page tab.

   Optionally, click **Create your own style** on the Tab Style Selector dialog to create a custom tab style if your org has access to the Documents tab. To create your own tab style:

   a. Click the **Color** lookup icon to display the color selection dialog and click a color to select it.

   b. Click **Insert an Image**, select the document folder, and select the image you want to use.

      Alternatively, click **Search in Documents**, enter a search term, and click **Go!** to find a document file name that includes your search term.

      Note: This dialog only lists files in document folders that are under 20 KB and have the Externally Available checkbox selected in the document property settings. If the document used for the icon is later deleted, Salesforce replaces it with a default multicolor block icon ( ).

   c. Select a file and click **OK**. The New Custom Tab wizard reappears.

6. Enter a description of the tab, if desired, and click **Next**.

7. Choose the user profiles for which the new custom tab will be available.

   Note: In Salesforce Classic, the Default On and Default Off options for Lightning Page tabs don't work the same way as for other custom tabs. The Lightning Page menu item appears for the selected profiles in Salesforce1 whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning Page for the selected profiles.

8. Click **Save**.

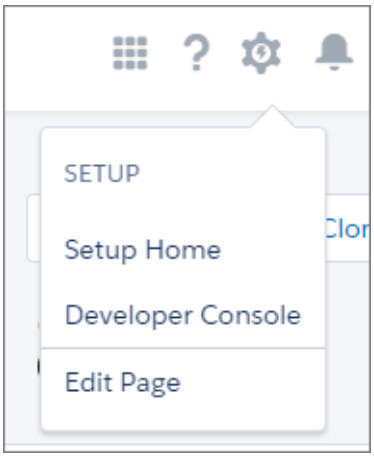
Configure Lightning Experience Record Pages

Use the Lightning App Builder to add, remove, or reorder components on a record page in Lightning Experience to give users a customized view for each object's records.

You can create a record page for Lightning Experience in two ways.

1. Edit a record page in Lightning Experience.

   a. From the Setup menu on a record page, select **Edit Page**.



EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

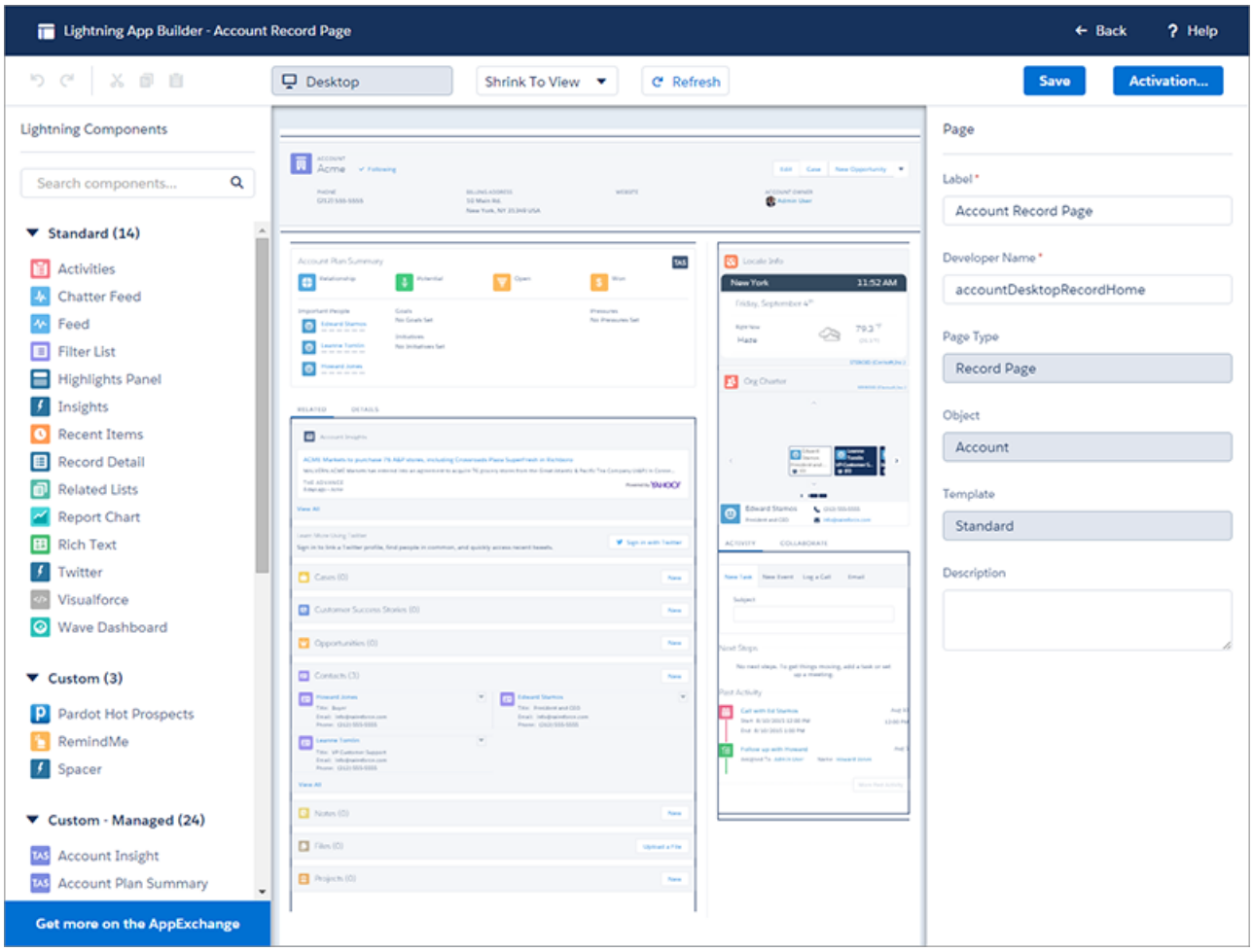
USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder
- "Customize Application"

To view Lightning Pages in the Lightning App Builder
- "View Setup and Configuration"

When you select Edit Page for the first time, Salesforce makes a copy of the standard page. This copy is what you then edit in the Lightning App Builder. If a customized page already exists and is active, selecting Edit Page opens that page to edit.

2. Create a new page from the Lightning App Builder list page.
   a. From Setup, enter *App Builder* in the Quick Find box, then select **Lightning App Builder**.
   b. Click **New**.
   c. Select **Record Page**.
   d. Choose a page template, and then complete the steps in the new page creation wizard.
3. In the Lightning App Builder, add, edit, or remove components to change the page's layout. Reorder components by dragging them around the canvas.



4. In the page properties, give your customized page a unique, descriptive label.

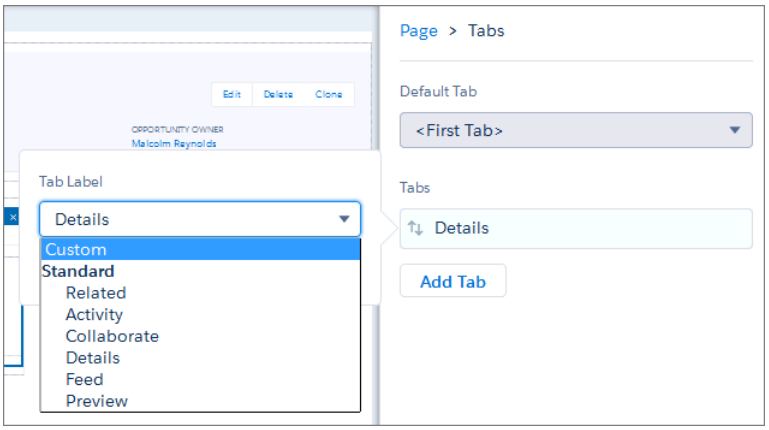
   To get to the page properties, click **Page** from the breadcrumb at the top of the properties pane.
5. Save your page.

   Hang on, you're not done yet! To make your customized record home page available to your Lightning Experience users, you must activate it. You can activate the page from the Save dialog when you save it for the first time, or later using the Activation button.

Customize Tabs on Lightning Experience Record Pages Using the Lightning App Builder

With the Lightning App Builder, you can create, update, delete, and change the order of tabs and tab sets on record pages in Lightning Experience. Configure the tabs that your users see, name them whatever you like, and add a custom set of components to each tab.

1. From the Lightning App Builder list page in Setup, open an existing record page or create a new one.
2. Add a Tabs component to the page.
3. In the Tabs component properties, click **Add Tab**.
   A Details tab is added.
4. Customize the Details tab by clicking it in the properties pane. You can select a different standard label or click **Custom** and enter the tab name you want.



EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

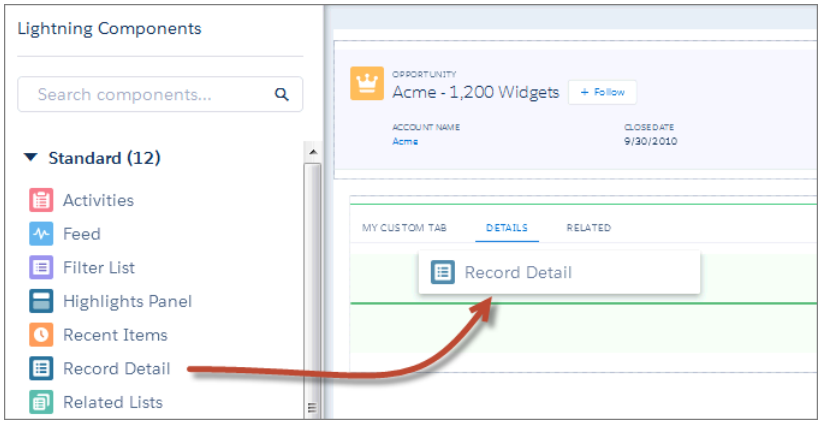
USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder
- "Customize Application"

To view Lightning Pages in the Lightning App Builder
- "View Setup and Configuration"

Note: Custom tab labels in the Tabs component—including those installed from packages—aren't translated. For example, if you create a custom "Goals" tab in English, then view the page as a user whose language is set to French, the tab still displays as "Goals".

5. To add your first component to a tab, select the tab on the canvas and then drop the component directly below it.



6. Reorder tabs in the Tabs properties pane by dragging and dropping them into position. You can't drag and drop individual tabs on the canvas.

Activate Lightning Experience Record Pages

Is your custom record page ready for your users? Use the Activation function inside the Lightning App Builder to get it out to them. You have two activation options. You can make your custom record page the default record page for all users or assign it to specific Lightning apps. Assigning the page to a Lightning app gives your users access to a record page customized for the app they're working in.

1. Create a record page or open an existing one in the Lightning App Builder.
2. If you opened a record page that is ready for your users and doesn't need editing, click **Activation**.
3. If you created a page or opened a page that needs adjustment, make the necessary changes, click **Save**, and then click **Activate**.
   You have two options for activation.
   - Make the page the default record page. Every user sees it unless they're in an app that already has an assigned custom record page.
   - Assign the page to specific Lightning apps.
4. To make the record page the default, select that option, review the assignments, and then click **Save**.

   Tip: If you no longer want this page to be the default, reopen the page in the Lightning App Builder. Redo the activation process, but this time select the option to remove it as the default.
5. To assign the page to specific apps, select that option and select which Lightning apps you want the page to display in. Review the assignments, and then click **Save**.

   You can only assign a custom record page to Lightning apps. You can see which record pages are activated for which Lightning app in the object's Lightning Record Pages related list in the Object Manager.

   Note: If you activate a page and then return to make changes, you don't have to activate it again. Clicking **Save** after you make your edits pushes the changes to your users.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder
- "Customize Application"

To view Lightning Pages in the Lightning App Builder
- "View Setup and Configuration"

Lightning Page Considerations and Limitations

Here are some things to keep in mind when working with Lightning Pages.

General Considerations

- If you're curious about a component, click it to see more information in the properties pane.
- You can create and edit a record page even if you don't have permission to access the object that the page is associated with. You can add, remove, delete, and reconfigure components on the page, but you don't see any of the content for the components that are based on that object.
- When you open a Lightning Experience record page in Lightning App Builder, you see only the components in the palette that are available for the object tied to that page.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Lightning Page Tabs

- In Salesforce Classic, Lightning Page tabs don't show up on the All Tabs page when you click the plus icon ( ) that appears to the right of your current tabs. Nor do Lightning Page tabs show up in the Available Tabs list when you customize the tabs for your apps.
- In Salesforce Classic, the Default On and Default Off options for Lightning Page tabs don't work the same way as for other custom tabs. The Lightning Page menu item appears for the selected profiles in Salesforce1 whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning Page for the selected profiles.

Limits and Limitations

- You can't assign record pages that you create with the Lightning App Builder to different profiles or record types. However, you can assign them to different custom apps.
- You can place up to 25 tabs in a Tabs component.
- A Lightning Page region or tab can contain up to 25 components.
- You can have both a Visualforce or s-control override and a Lightning Page override set for the View action at the same time on the same object. In Lightning Experience, the Lightning Page override displays. In Salesforce Classic, the Visualforce or s-control override displays.
- Drop-down menus in the Lightning App Builder can display up to 200 items. Enter a few characters, and all available matches display as you type.
- You can't upload a record page to a managed package.
- You can't edit person account pages because they're read-only.
- Custom tab labels in the Tabs component—including those installed from packages—aren't translated. For example, if you create a custom "Goals" tab in English, then view the page as a user whose language is set to French, the tab still displays as "Goals".

SEE ALSO:

Considerations and Limitations for Flows in Lightning Pages (Beta)

Considerations and Limitations for Flows in Lightning Pages (Beta)

Here are some things to keep in mind when you add a flow component to a Lightning Page.

Note: This release contains a beta version of the Flow component for Lightning Pages, which means it's a high-quality feature with known limitations. The Flow component isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only on the basis of generally available products and features. You can provide feedback and suggestions for the Flow component for Lightning Pages in the IdeaExchange.

Lightning Pages always use Lightning runtime, so also review the Lightning Runtime Beta Limitations.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Running Flows from a Lightning Page

When a user opens a Lightning Page with a flow component in it, the flow runs when the page loads. We recommend making sure that the flow doesn't do anything – such as create or delete records – before the first screen.

Updating Versions of the Referenced Flow

After you add a flow to a Lightning page, be careful about activating new versions of that flow. If the flow's input variables change–such as the name or data type of the variable–the flow component can become invalid. As a workaround, remove that flow component and replace it with a new one.

Input Variable Limitations

- These variables aren't supported:
  - Collection variables
  - sObject variables
  - sObject collection variables
  - Variables of type Picklist or Multi-Select Picklist
- The component supports only manually entered values for input variables.
- Text input variables accept a maximum length of 4,000 characters.
- The property editor truncates values for number input variables after 3 decimal places, even if the variable's scale is higher than 3. However, the flow uses the non-truncated value at run time.

  For example, if you enter 3.12345, the property editor later displays 3.123, but at run time the flow uses 3.12345.
- If your flow includes an input variable with the following settings, and you reference that flow from a Record Page, the record's ID is automatically passed into that variable at run time.

Field	Value
Unique Name	recordId
Data Type	Text
Input/Output Type	Input and Output OR Input Only

Deployment Limitations

- If a Lightning Page or FlexiPage contains a flow component, we don't support:
  - Creating packages that include the Lightning Page
  - Copying a sandbox that includes the Lightning Page or FlexiPage
  - Creating trialforce templates that include the FlexiPage
- If you use the Metadata API to deploy a FlexiPage that includes a flow component, the referenced flow must already exist. If the flow doesn't already exist, deploy the flow first and then deploy the FlexiPage.

SEE ALSO:

Limits and Considerations for Visual Workflow

Lightning Pages

Lightning Page Considerations and Limitations

Considerations for Two-Column Flows (Beta)

Resources for the Point & Click Administrator

In addition to online help, Salesforce creates video demos, guides, and tip sheets to help you learn about our features and successfully administer Salesforce.

Platform and Apps

	For End Users	For Admins
Guides and Tip Sheets		
Building Custom Objects, Tabs, and Related Lists		✔
Implementing State and Country Picklists		✔
Using Multiple Business Processes		✔
Useful Workflow Rules		✔
Useful Approval Processes		✔
Cloud Flow Designer Workbook		✔
Visual Workflow Guide		✔
Force.com Sites Implementation Guide		✔
Organization Sync Implementation Guide		✔
Videos		
How to Change the Look and Feel of Salesforce for Your Company		✔
Creating a Workflow Rule		✔
Visual Workflow Cloud Flow Designer Overview		✔
Building a Simple Flow		✔
Troubleshooting Workflow		✔
Avoid Unwanted Actions in Your Process		✔
Creating Custom Links and Buttons		✔
How Organization Sync Works		✔
Who Sees What: Record Types		✔

Custom Fields

	For End Users	For Admins
Videos		

	For End Users	For Admins
How to Create a Custom Field in Salesforce		✓
Custom Fields: Picklists		✓

Formulas

	For End Users	For Admins
Guides and Tip Sheets		
Formulas Quick Reference Guide		✓
Useful Formula Fields		✓
Tips for Reducing Formula Size		✓
Using Date and Date/Time in Formulas		✓
Useful Validation Rules		✓
Videos		
Getting Started With Formulas		✓
How to Build an Image Formula		✓
Formulas: Tips and Gotchas		✓

INDEX

A

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W