The Lightning Experience Implementation Guide

For Orgs Transitioning from Salesforce Classic

Version 11, Winter '20

Written by
Michelle Chapman-Thurber
Tammy Rahn
Michael Alderete

With contributions by
Jennifer Crabtree
Cate deHeer
Brian Donnelly
Chris Duarte
Chalon Emmons
Justine Heritage
Michael Hoban
Dave Jacowitz
Erin Tidwell
Lance Santin
Leah Scampoli
Jillian West

This guide is for Salesforce admins and developers who are transitioning Salesforce Classic orgs to Lightning Experience. We take you through a virtual tour of the Lightning Experience features and functionality that we recommend implementing for your users. Then get suggestions on how to customize Lightning Experience, including options for adjusting existing customizations. Finally, we've included information about how Lightning Experience changes the way you develop applications, as well as detailed information about Visualforce in Lightning Experience.

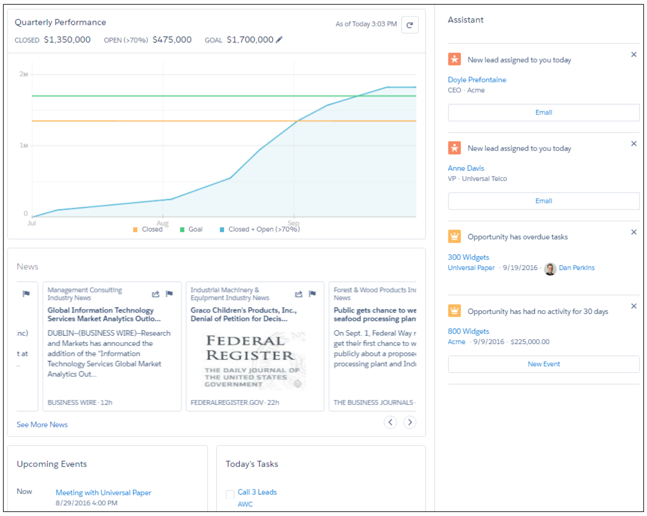
The Lightning Experience Implementation Guide

CONTENTS

CHAPTER 1 The Time for Lightning Experience Is Now

In this chapter ...

Salesforce Lightning is the next generation of Salesforce. Lightning includes Lightning Experience (our reimagined user experience), the powerful Lightning Platform for fast and easy app development, and the AppExchange ecosystem for ready-to-install solutions. The new user interface is designed with a sales- and service-centric mindset that helps teams work more naturally and productively. When you harness the power of Lightning, you can transform your business and benefit everyone at your company — from execs and business users to admins and developers.



Quarterly Performance

As of Today 3:03 PM

CLOSED $1,350,000 OPEN (>70%) $475,000 GOAL $1,700,000

News

Management Consulting Industry News

Global Information Technology Services Market Analytics Outlo...

DUBLIN--(BUSINESS WIRE)--Research and Markets has announced the addition of the "Information Technology Services Global Market Analytics Out...

BUSINESS WIRE · 12h

Industrial Machinery & Equipment Industry News

Graco Children's Products, Inc., Denial of Petition for Decis...

FEDERALREGISTER.GOV · 22h

Forest & Wood Products Inc News

Public gets chance to we seafood processing plan

On Sept. 1, Federal Way r get their first chance to w publicly about a propose processing plant and Indi

THE BUSINESS JOURNALS ·

See More News

Upcoming Events

Now Meeting with Universal Paper 8/29/2016 4:00 PM

Today's Tasks

Call 3 Leads AWC

Assistant

New lead assigned to you today

Doyle Prefontaine CEO · Acme

Email

New lead assigned to you today

Anne Davis VP · Universal Telco

Email

Opportunity has overdue tasks

300 Widgets Universal Paper · 9/19/2016 · Dan Perkins

Opportunity has had no activity for 30 days

800 Widgets Acme · 9/9/2016 · $225,000.00

New Event

The new Home Page in Lightning Experience

Before we go on, let's play a round of Debunk the Myth. Many customers automatically assume that when they say hello to Lightning Experience, they must say goodbye to Salesforce Classic. But guess what?

You and your users can still use Salesforce Classic as if nothing has changed. Everyone gets a link so they can move easily between the new and classic interfaces as needed. It's good to have options, yeah?

Okay, cool, Salesforce Classic isn't going away yet. *But*. . . All new Salesforce innovation is in Lightning Experience only. So clearly the new interface is the place to be. How do you get there, you're wondering?

Use this guide for help with implementing Lightning Experience so it meets the needs of your business and users. You'll get a handle on the Lightning Experience features that create a more productive experience. And you'll learn about recommended customization steps, including making sure your existing customizations in Salesforce Classic work in Lightning Experience.

For help with the overall transition from Salesforce Classic to Lightning Experience, including how to evaluate your readiness and develop a rollout plan, see *How to Transition to Lightning Experience* in Salesforce Help.

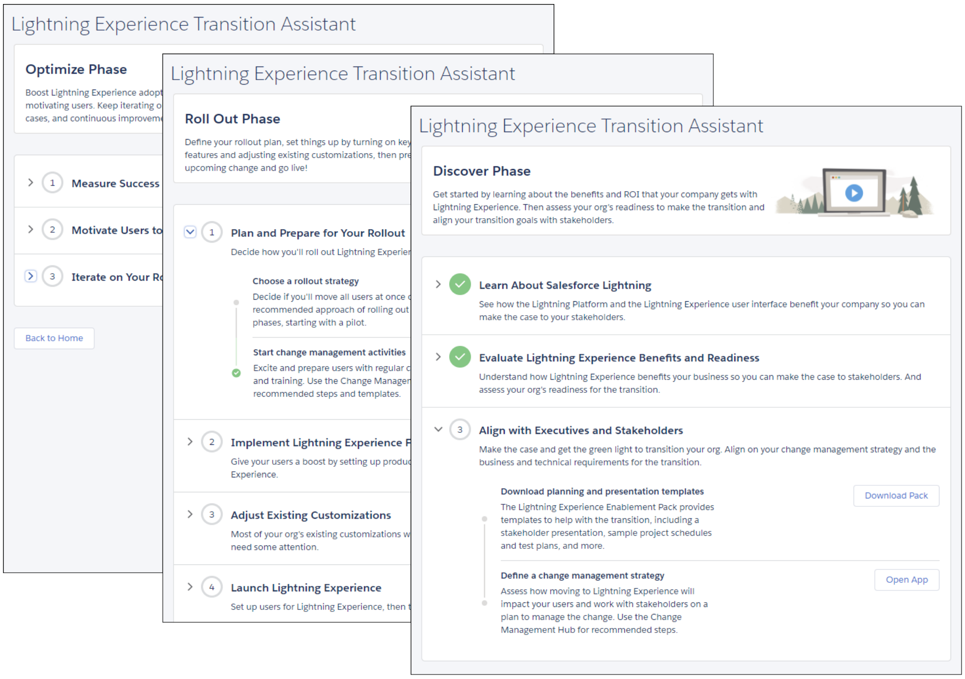
A Framework for Your Lightning Experience Transition

Depending on the size of your organization, you could be working with a Change Management department or have a project manager assigned to the transition. Or perhaps you're the one charged with organizing and executing things from start to finish. However the work gets divvied up, the Lightning Experience transition framework puts you on the right path!

Based on our experiences and those of the thousands of customers who've already moved to Lightning Experience, we defined a framework that outlines the structure and best practices you can follow for your own journey. For easier manageability, we recommend approaching your transition in three phases. The phases break down into a few stages, each with a set of recommended activities.

We'll learn about these activities in the rest of this module. But you'll be happy to know that the framework isn't purely conceptual. It comes to life in the Lightning Experience Transition Assistant, a tool that guides you through each phase, stage, and activity in your transition.

The Transition Assistant is available from Setup. From Salesforce Classic, take a peek by clicking **Get Started** in the Lightning Experience Transition Assistant tile at the top of the menu.



Lightning Experience Transition Assistant

Optimize Phase

Boost Lightning Experience adopt
motivating users. Keep iterating o
cases, and continuous improveme

1 Measure Success
2 Motivate Users to
3 Iterate on Your Ro

Back to Home

Lightning Experience Transition Assistant

Roll Out Phase

Define your rollout plan, set things up by turning on key
features and adjusting existing customizations, then pre
upcoming change and go live!

1 **Plan and Prepare for Your Rollout**
Decide how you'll roll out Lightning Experien

Choose a rollout strategy
Decide if you'll move all users at once o
recommended approach of rolling out
phases, starting with a pilot.

Start change management activities
Excite and prepare users with regular c
and training. Use the Change Manage
recommended steps and templates.

2 **Implement Lightning Experience F**
Give your users a boost by setting up produc
Experience.

3 **Adjust Existing Customizations**
Most of your org's existing customizations w
need some attention.

4 **Launch Lightning Experience**
Set up users for Lightning Experience, then t

Lightning Experience Transition Assistant

Discover Phase

Get started by learning about the benefits and ROI that your company gets with Lightning Experience. Then assess your org's readiness to make the transition and align your transition goals with stakeholders.

✓ **Learn About Salesforce Lightning**
See how the Lightning Platform and the Lightning Experience user interface benefit your company so you can make the case to your stakeholders.

✓ **Evaluate Lightning Experience Benefits and Readiness**
Understand how Lightning Experience benefits your business so you can make the case to stakeholders. And assess your org's readiness for the transition.

3 **Align with Executives and Stakeholders**
Make the case and get the green light to transition your org. Align on your change management strategy and the business and technical requirements for the transition.

Download planning and presentation templates
The Lightning Experience Enablement Pack provides templates to help with the transition, including a stakeholder presentation, sample project schedules and test plans, and more.

Download Pack

Define a change management strategy
Assess how moving to Lightning Experience will impact your users and work with stakeholders on a plan to manage the change. Use the Change Management Hub for recommended steps.

Open App

Don’t worry, moving to Lightning Experience doesn’t have to be a massive, lengthy effort. In fact, many customers begin moving users within a month or two of starting their transitions. We recommend taking an iterative, agile approach where you break up the rollout into several groups, starting with a pilot, so you don’t have to do everything at once and can learn and refine as you go. And you only need to use the parts of the framework that make sense for your company.

Lightning Experience Enablement Pack

To help jump-start your move to Lightning Experience, we provide an Enablement Pack that’s chock-full of rollout resources and customizable templates. Use the pack to prepare for and complete a successful Lightning Experience transition. We’ll recommend specific resources in the Enablement Pack as we go. To get a head start, you can download it now.

CHAPTER 2 What Makes Lightning Experience So Special

In this chapter ...

With all its new features and redesigned pages, there are many key benefits to using Lightning Experience. We won't cover everything here, but for each area of the product, we'll highlight some of the neatest things, including:

- Efficient navigation and the ability to switch between custom-branded apps
- New record layouts that focus on what you can do instead of what you can view
- Turbocharged list views that let you easily filter and visualize your data
- Quick access to productivity tools like Notes and Recent Items in the utility bar
- Beautiful dashboards with components that span both columns and rows
- Sleek report views that you can filter quickly to see the data that's most important to you

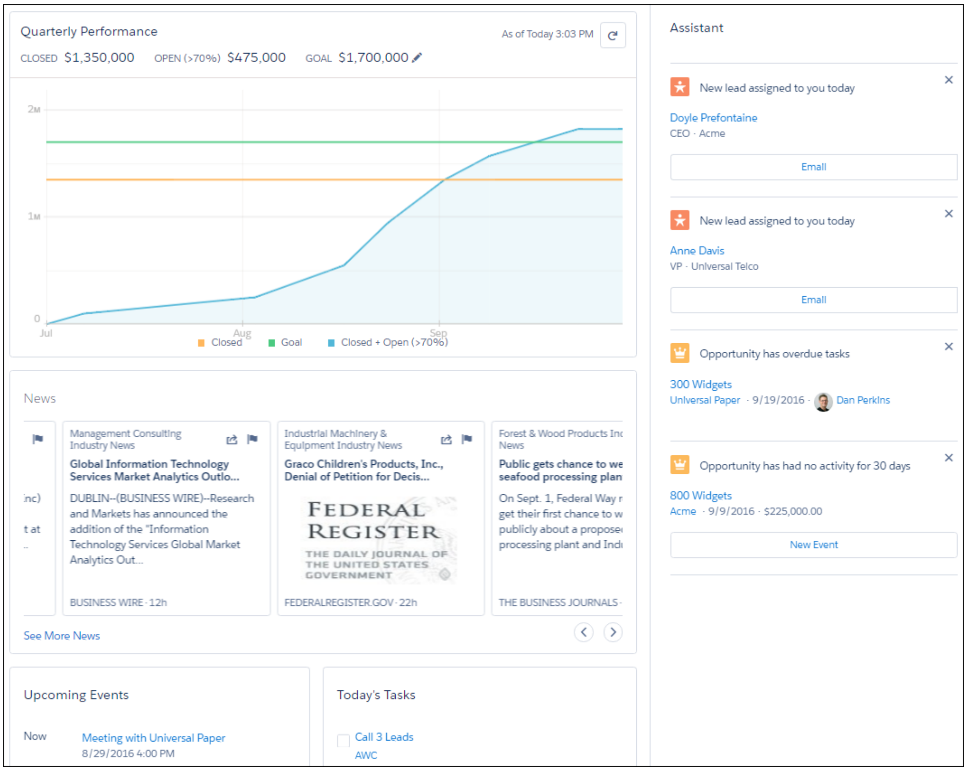
Take a Tour of Lightning Experience

Why is Lightning Experience so special? We'll show you!

Home

We've re-imagined the way you start your day with a brand new, intelligent Home page. Now your sales reps can monitor their performance to goal and get insights on their key accounts. Plus, we've added an Assistant, which is your users' action list of things to do. You can also use the Lightning App Builder to create custom Home pages that appear for different profiles.

- Start your day fast with a customizable, intelligent page.
- Monitor how close you are to crushing your numbers with the Performance Chart.
- See relevant, timely news articles about customers, partners, and competitors with News.
- See upcoming meetings and tasks due today.
- Identify key issues to work on each day with the Assistant.
- Focus your selling activities on your Top Deals.



News

Management Consulting Industry News
Global Information Technology Services Market Analytics Outlo...
DUBLIN--(BUSINESS WIRE)--Research and Markets has announced the addition of the "Information Technology Services Global Market Analytics Out...
BUSINESS WIRE · 12h

Industrial Machinery & Equipment Industry News
Graco Children's Products, Inc., Denial of Petition for Decis...
FEDERAL REGISTER THE DAILY JOURNAL OF THE UNITED STATES GOVERNMENT
FEDERALREGISTER.GOV · 22h

Forest & Wood Products Inc News
Public gets chance to we seafood processing plan
On Sept. 1, Federal Way r get their first chance to w publicly about a propose processing plant and Indi
THE BUSINESS JOURNALS ·

See More News

Upcoming Events
Now Meeting with Universal Paper
8/29/2016 4:00 PM

Today's Tasks
Call 3 Leads
AWC

Assistant

New lead assigned to you today
Doyle Prefontaine
CEO · Acme
Email

New lead assigned to you today
Anne Davis
VP · Universal Telco
Email

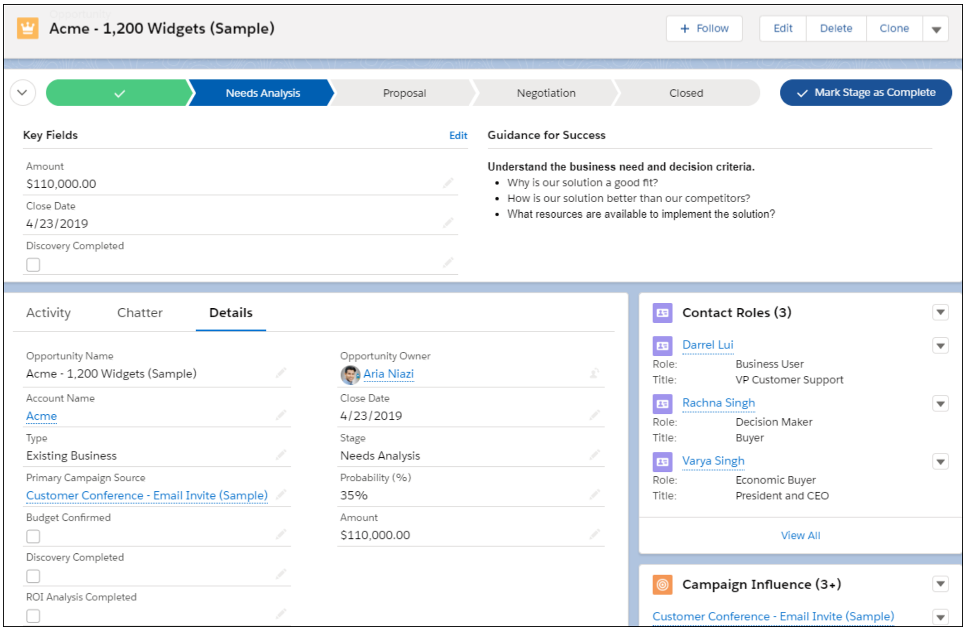
Opportunity has overdue tasks
300 Widgets
Universal Paper · 9/19/2016 · Dan Perkins

Opportunity has had no activity for 30 days
800 Widgets
Acme · 9/9/2016 · $225,000.00
New Event

Opportunity Workspace

We've taken your sales process and put it into an action-optimized workspace, designed to help your sales reps work their deals faster and smarter. You can customize coaching scripts for each step in the sales process, create records quickly with fewer clicks, and ultimately close deals faster.

- Showcase key record details in the new highlights panel at the top of the page.
- Use the handy composer to quickly log calls, create tasks, send emails, and more.
- Get key coaching details with a customizable Path to support your sales process.
- See a wealth of related information on hover using quick view, without ever leaving the opportunity page.
- Add related records—like contacts—in context with minimal clicks.
- View relevant, timely news about the account the opportunity's associated with.



Opportunity
Acme - 1,200 Widgets (Sample)

Key Fields

Amount
$110,000.00
Close Date
4/23/2019
Discovery Completed

Guidance for Success

Understand the business need and decision criteria.
- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

Opportunity Name: Acme - 1,200 Widgets (Sample)	Opportunity Owner: Aria Niazi
Account Name: Acme	Close Date: 4/23/2019
Type: Existing Business	Stage: Needs Analysis
Primary Campaign Source: Customer Conference - Email Invite (Sample)	Probability (%): 35%
Budget Confirmed:	Amount: $110,000.00
Discovery Completed:	
ROI Analysis Completed:	

Contact Roles (3)

Darrel Lui
Role: Business User
Title: VP Customer Support

Rachna Singh
Role: Decision Maker
Title: Buyer

Varya Singh
Role: Economic Buyer
Title: President and CEO

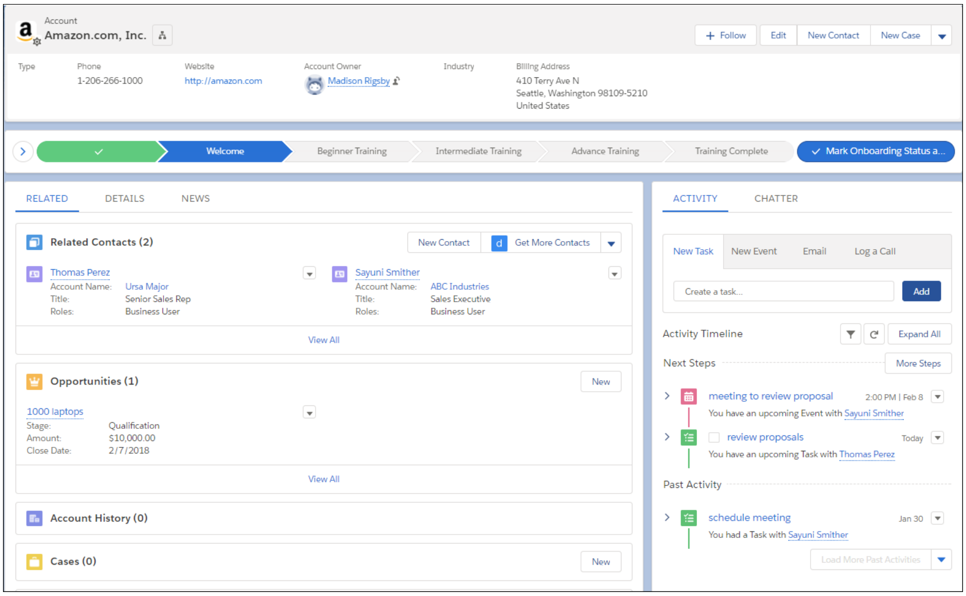
Campaign Influence (3+)

Customer Conference - Email Invite (Sample)

Accounts and Contacts

Like with opportunities (and leads), we've optimized the layout for accounts and contacts, organizing the content by their primary use case: reference. Now your sales and service reps can find information and gather information at a glance.

- Locate important data efficiently with the redesigned Lightning Experience page layout.
- Get key coaching details on account records with a customizable Path to support your sales process.
- Get the latest news for your customers with News.
- Work smarter and keep your data clean with field-level duplicate matching.
- Review past and upcoming activities at a glance.


Account
Amazon.com, Inc.

+ Follow | Edit | New Contact | New Case

Type	Phone	Website	Account Owner	Industry	Billing Address
	1-206-266-1000	http://amazon.com	Madison Rigsby		410 Terry Ave N Seattle, Washington 98109-5210 United States

Welcome | Beginner Training | Intermediate Training | Advance Training | Training Complete | ✓ Mark Onboarding Status a...

RELATED | DETAILS | NEWS

Related Contacts (2) — New Contact | Get More Contacts

Thomas Perez		Sayuni Smither	
Account Name:	Ursa Major	Account Name:	ABC Industries
Title:	Senior Sales Rep	Title:	Sales Executive
Roles:	Business User	Roles:	Business User

View All

Opportunities (1) — New

1000 laptops

Stage:	Qualification
Amount:	$10,000.00
Close Date:	2/7/2018

View All

Account History (0)

Cases (0) — New

ACTIVITY | CHATTER

New Task | New Event | Email | Log a Call

Create a task... — Add

Activity Timeline — Expand All

Next Steps — More Steps

meeting to review proposal — 2:00 PM | Feb 8
You have an upcoming Event with Sayuni Smither

review proposals — Today
You have an upcoming Task with Thomas Perez

Past Activity

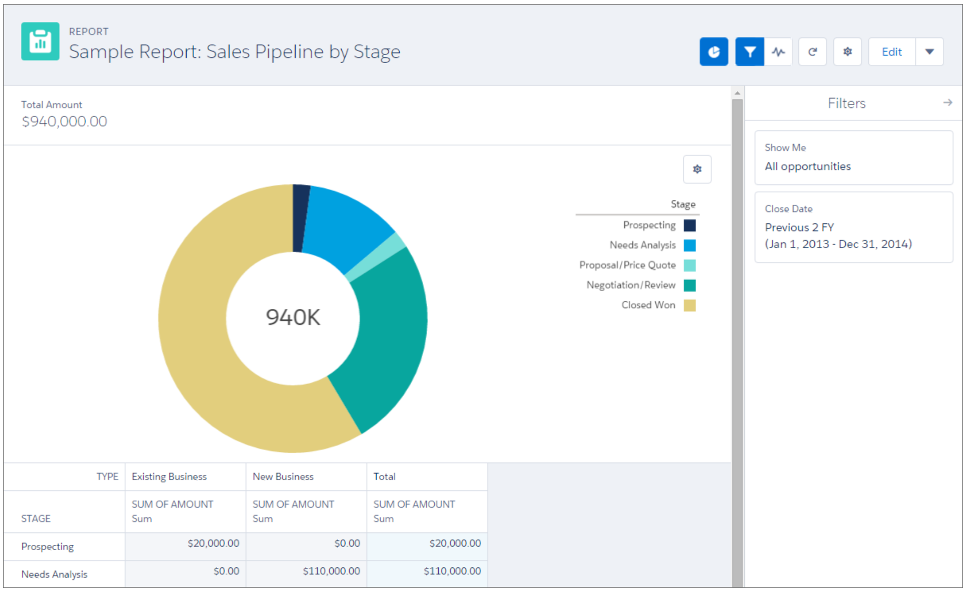
schedule meeting — Jan 30
You had a Task with Sayuni Smither

Load More Past Activities

Reports and Dashboards

Your users will love the ability to create their own filters on reports. You will appreciate the updated dashboard editor, which features components that span both columns and rows.

- Create filters while viewing a report.
- Make visually awesome dashboards with more than three columns.
- Transition easily from Salesforce Classic to Lightning Experience with reports and dashboards that are automatically viewable in the new interface. And they inherit all permissions and sharing settings that were defined in Salesforce Classic.

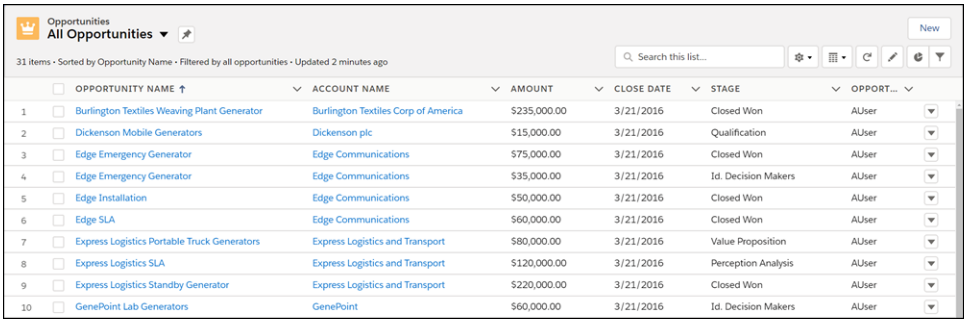


TYPE	Existing Business	New Business	Total
STAGE	SUM OF AMOUNT Sum	SUM OF AMOUNT Sum	SUM OF AMOUNT Sum
Prospecting	$20,000.00	$0.00	$20,000.00
Needs Analysis	$0.00	$110,000.00	$110,000.00

List Views

Now your sales and service reps can visualize any list view graphically with a handy chart, or easily apply filters to narrow the results.

- Visualize your data in seconds with list view charts, and quickly create filters to slice your data how you want.
- Create list views by using Lightning Experience's intuitive filters panel.
- Use type-ahead search to find a favorite list view fast.
- Automatically open your list views created in Salesforce Classic from Lightning Experience.
- Use inline editing to make quick changes to records in a list view.
- Find the data you need in a snap with the list view search bar.
- Share your list views with user groups in your org.
- Pin a list view to make it load as your default list.



Opportunities
All Opportunities

New

31 items • Sorted by Opportunity Name • Filtered by all opportunities • Updated 2 minutes ago

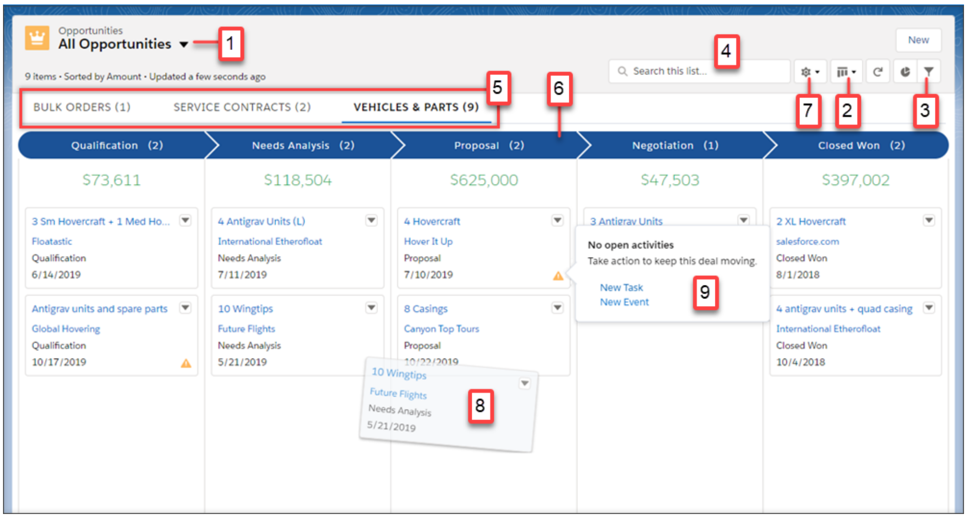
Search this list...

	OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORT...
1	Burlington Textiles Weaving Plant Generator	Burlington Textiles Corp of America	$235,000.00	3/21/2016	Closed Won	AUser
2	Dickenson Mobile Generators	Dickenson plc	$15,000.00	3/21/2016	Qualification	AUser
3	Edge Emergency Generator	Edge Communications	$75,000.00	3/21/2016	Closed Won	AUser
4	Edge Emergency Generator	Edge Communications	$35,000.00	3/21/2016	Id. Decision Makers	AUser
5	Edge Installation	Edge Communications	$50,000.00	3/21/2016	Closed Won	AUser
6	Edge SLA	Edge Communications	$60,000.00	3/21/2016	Closed Won	AUser
7	Express Logistics Portable Truck Generators	Express Logistics and Transport	$80,000.00	3/21/2016	Value Proposition	AUser
8	Express Logistics SLA	Express Logistics and Transport	$120,000.00	3/21/2016	Perception Analysis	AUser
9	Express Logistics Standby Generator	Express Logistics and Transport	$220,000.00	3/21/2016	Closed Won	AUser
10	GenePoint Lab Generators	GenePoint	$60,000.00	3/21/2016	Id. Decision Makers	AUser

Kanban View

The Kanban view organizes a set of records into columns to track your work at a glance. To update a record's status, drag it into a different column. You can configure the board by selecting what fields columns and summaries are based on. And, get personalized alerts on key opportunities in flight.

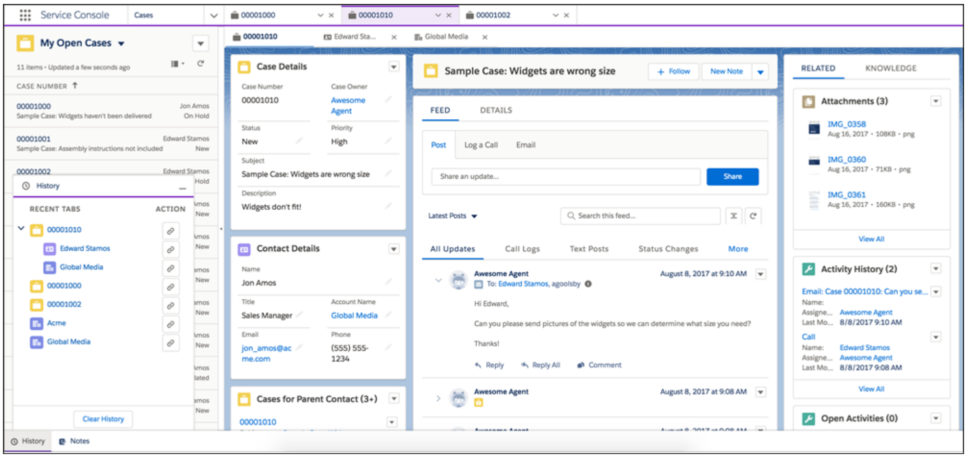
- Visualize your work at each stage or status
- Move records between columns using drag and drop functionality
- Configure columns and summary fields on the fly
- Edit or delete records to keep them up to date
- Quickly create filters to slice your data how you want
- For opportunities, get alerts to notify you when action is needed on a key deal



1. The records in the Kanban view are based on the selected list view.
2. Easily toggle between the list view grid view and the Kanban view.
3. Filter your records to view a particular subset of your records.
4. Search for records within the current view.
5. Select which record type to view.
6. Columns are created based on the grouping field.
7. Change how columns are organized and summarized using Kanban settings.
8. Quickly move a record to a different column by dragging the card.
9. For opportunities, alerts tell how to keep a deal on track, for example, create a task or event.

Console Apps

Minimize clicks and scrolling for your sales and service reps so they can quickly find, update, and create records. A Lightning console app is a tab-based workspace that lets users manage multiple records on a single screen. Take advantage of the standard Lightning console apps for Sales and Service to support your fast-paced environment.



00001000	Jon Amos	On Hold
Sample Case: Widgets haven't been delivered		
00001001	Edward Stamos	New
Sample Case: Assembly instructions not included		
00001002	Edward Stamos	Hold

Recent tabs: 00001010, Edward Stamos, Global Media, 00001000, 00001002, Acme, Global Media

Case Number: 00001010 — Case Owner: Awesome Agent
Status: New — Priority: High
Subject: Sample Case: Widgets are wrong size
Description: Widgets don't fit!

Name: Jon Amos
Title: Sales Manager — Account Name: Global Media
Email: jon_amos@acme.com — Phone: (555) 555-1234

Awesome Agent — August 8, 2017 at 9:10 AM
To: Edward Stamos, agoolsby
Hi Edward,
Can you please send pictures of the widgets so we can determine what size you need?
Thanks!

Awesome Agent — August 8, 2017 at 9:08 AM

IMG_0358 — Aug 16, 2017 • 108KB • png
IMG_0360 — Aug 16, 2017 • 71KB • png
IMG_0361 — Aug 16, 2017 • 160KB • png

Email: Case 00001010: Can you se...
Assigne... Awesome Agent
Last Mo... 8/8/2017 9:10 AM
Call
Name: Edward Stamos
Assigne... Awesome Agent
Last Mo... 8/8/2017 9:08 AM

Other Highlights

- Make your users more efficient and allow them to switch between apps that you can brand and customize. Create records and access recent records and lists for certain items directly from the navigation bar.
- See open tasks, planned meetings, and accomplishments in the activity timeline on activity-enabled objects. Quickly log calls, create tasks, send emails, and more from the handy composer.
- Find your tasks on a new page for tasks, including the new master-detail view, which lets you see a single task and your entire list of tasks side by side.
- Use phone features without ever leaving Salesforce. Make and receive calls, add call notes, and log call information with ease. *Available for an extra cost.*
- Send email through Gmail or Office 365 accounts with your Salesforce email. And see the email messages you've sent in your Gmail or Office 365 Sent Items folder for flawless integration.
- Take better notes with auto-save, rich text capabilities, inline images, and versioning. Relate a note to multiple records and share notes with teammates or Chatter groups.
- Enjoy a richer file preview experience that doesn't require Adobe Flash Player.
- Find records faster with improved global search and lookup search. Search results include instant results, top results, and recent records.

As you can see, there's a lot of rad stuff that you get with Lightning Experience. Next, we'll take you through the key differences between Lightning Experience and Salesforce Classic, to help you figure out how you'll make the transition.

How Different Personas Benefit from Lightning Experience

So now you have a good sense of some of the exciting features that make Lightning Experience special. But at the end of the day, how do these features and other aspects of Lightning Experience actually benefit you and your users?

The short answer: Lightning Experience makes people more productive, no matter what role they play. And scout's honor, that's not just hyperbole. Customers who've made the move to Lightning Experience are seeing sizable increases in efficiency, including less time spent managing pipelines, faster win rates, better agent productivity, and faster deployments.

Let's look at the productivity benefits for each Salesforce persona.

Sales Users

Lightning Experience is a streamlined, action-oriented environment that promotes an efficient sales workflow. In every context, pages focus on the most important information first. Workspaces make your sales process clear, supported by best practices at every step. Separate tabs for details, related information, and collaboration bring order to pages and reduce the amount of scrolling to find things. Actions like taking notes, making calls, and sending emails are right where they're needed, in context. And baked-in intelligence, from news and insights to Salesforce Einstein, tees up teams to close more deals in less time.

Service Users

With Service Cloud for Lightning Experience, service agents are on the case resolution fast track. The Lightning Service Console is a high-productivity workspace with a single view of every customer so service agents don't have to waste time toggling between multiple apps. The console guides agents through your case management process, and makes every step faster to complete with efficiencies like split view, drag-and-drop attachments, and case detail hovers. Add Lightning Knowledge to the mix and agents have the tools to find and share the right information, attach articles, and get to closed cases faster.

Admins

With the Lightning framework, admins can create and customize pages faster, without having to start from scratch each time. Don't let the term *framework* scare you off. Admins can work their magic using the framework's declarative tools that are available from Setup in Lightning Experience.

The building blocks that make up Lightning Experience are Lightning components. These components are reusable throughout the interface so it's easy to create just what's needed. We provide a bunch of pre-built components to get started, and an intuitive visual tool called Lightning App Builder. Admins can add, remove, and rearrange components on the home page or record pages with drag-and-drop ease. And beyond page layout decisions, admins can control how components behave, configuring the same component to use different properties in different locations...even on the same page. This flexibility significantly speeds up customization work.

Developers

The Lightning framework is a productivity boon for developers too. It's fast to create custom Lightning components using the building blocks we include in the framework. And that's just the start. The framework provides many other tools to help developers. For example, the Salesforce Lightning Design System is a complete styling library so developers don't have to write their own HTML or CSS. Lightning Data Service consolidates data access code in reusable components, eliminating the need to write controller code. That's just scratching the surface. But the added efficiency for developers is obvious: all this reduced complexity means that developers can code faster and spend less time on performance and quality testing.

CHAPTER 3 Navigation and Setup in Lightning Experience

In this chapter ...

Setup is where you make the magic happen. As a Salesforce admin or developer, you spend a lot of time using Setup. It's where you customize and configure your organization, support users, build functionality, and more.

One of the huge productivity upgrades that comes with the new Lightning Experience is the improved Setup. We've done a lot of usability testing and refactoring to revamp the Setup tree. We simplified it to have a logical and easy-to-navigate structure, using broad categories to make things more discoverable. In addition, child nodes are now in alphabetical order.

The navigation bar in Lightning Experience provides an efficient and consistent interface to navigate through your organization's various apps and items. Similar to Salesforce Classic, apps in Lightning Experience give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. However, apps in Lightning Experience take things to another level beyond apps in Salesforce Classic by letting you brand your apps with a custom color and logo. In Lightning Experience you can even include Lightning page tabs and a utility bar that allows instant access to productivity tools, like integrated voice, in the footer of Lightning Experience.

Ready to see them in action?

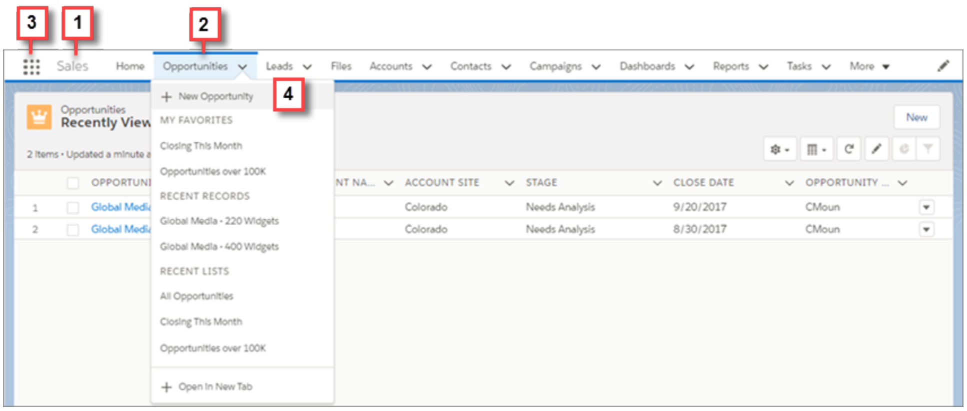
The Lightning Experience Navigation Bar

If you know Salesforce Classic, the Lightning Experience navigation model will feel like a familiar friend, only better.

Each Lightning app has a navigation bar at the top of the page, letting your users:

- Find what they need using item names for easy recognition
- Complete actions and access recent records and lists with a single click
- Personalize the navigation bar to suit the unique way they work

Think of the navigation bar as a container for a set of items and functionality. It's always there, but the items within it change based on the app you're using.



	OPPORTUNI	NT NA...	ACCOUNT SITE	STAGE	CLOSE DATE	OPPORTUNITY ...
1	Global Medi		Colorado	Needs Analysis	9/20/2017	CMoun
2	Global Medi		Colorado	Needs Analysis	8/30/2017	CMoun

- The app name (1) displays on the left side of the navigation bar and custom colors and branding (2) make each app unique and easy to identify.
- Your users can access other items and apps by clicking the App Launcher icon (3).
- Your users can create records and access recent records and lists directly from the navigation bar (4) for items like Opportunities.

What can you put in the Lightning app navigation bar?

If you're familiar with Salesforce Classic, you know that Classic apps can contain:

- 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 can contain everything on this list plus Lightning page tabs and utilities like Lightning Voice. If your org uses utility features, you can enable a utility bar in your app that allows instant access to productivity tools, like integrated voice, in the Lightning Experience footer.

As we mentioned, you and your users can still get to custom apps and objects via the App Launcher, which we'll look at next.

How can users personalize the navigation bar?

Personalized navigation in Lightning Experience is similar to customized tab sets in Salesforce Classic but better. In Lightning Experience, the navigation bar can contain more than just object-level items, like Accounts. You can add granular items, like a dashboard, list, or record. Let's look at the different ways users edit the navigation bar.

You can drag items on the navigation bar to change their order.

Note: The asterisk to the left of the item name indicates that it's a temporary tab. A temporary tab opens when you open an item that doesn't have a parent object already in the navigation bar. The temporary tab is removed from the nav bar when you close it, log out of Salesforce, or switch to a different app.

Want to make several changes to the navigation bar? Click the pencil icon on the navigation bar. From the enhanced editor:

- Reorder the items already in your navigation bar.
- Rename items you've added. A pencil icon is next to the items that you can rename.
- Add items to the navigation bar. Click **Add More Items**. Search through your favorites or all available items in your org, and choose what to add. After you make your selections, you can reorder or remove items before saving your changes. You can't delete items that your admin has specified for the app.

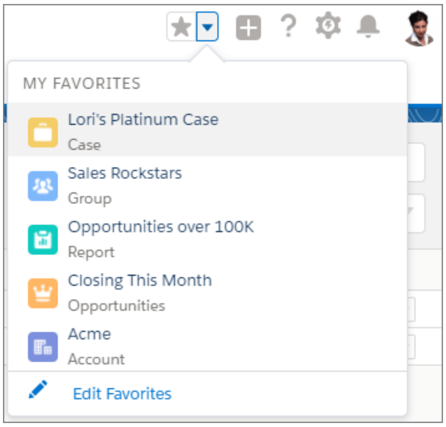
A few considerations to keep in mind.

- Help users get the most out of personalized navigation by upgrading your Classic apps to Lightning apps. Users can't personalize the navigation bar of Classic apps in Lightning Experience.
- Check what's in your apps. Users can't remove the items you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items.

- Items that you add to an app's navigation bar are added to the end of users' personalized navigation bars in the order that you added them.
- When you remove an item from an app, that item remains in your users' personalized navigation bars, and users can then delete it.
- If you don't want your users to personalize the navigation bar for a specific app, disable personalization. From Setup in Lightning Experience, go to the **App Manager**. For the desired app, select **App Options**. Select **Disable end user personalization of nav items in this app**.
- If you don't want your users to personalize the navigation bar for any app, disable personalization. From Setup, enter `User Interface` in the Quick Find box, then select **User Interface**. Select **Disable Navigation Bar Personalization in Lightning Experience**. Salesforce recommends disabling navigation personalization by app instead of for the entire org.
- Control if temporary tabs are created when users access items outside of the app. From Setup in Lightning Experience, go to the **App Manager**. Edit the desired app. On the **App Options** page, select **Disable temporary tabs for items outside of this app**.
- The tab's dropdown menu includes an option to create an item. For example, create an account by selecting **+ New Account** from the Account tab's dropdown menu. However, the New action doesn't appear if:
  - Users don't have the create permission for the object
  - The New action isn't part of the search layout for the object's list view

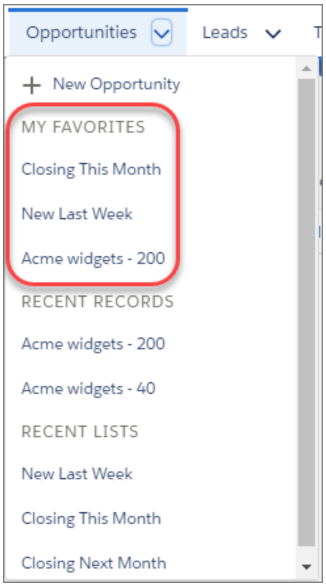
Fast Track Navigation in Lightning Experience with Favorites

Favorites are another great way for your users to boost productivity by personalizing how they navigate. Think of the favorites star at the top of the page in Lightning Experience as a constellation in the sky. It's always there, so your users can easily navigate to what they need no matter where they are.



Click the favorites star to add the current page—for example, opportunities closing this month—to your favorites. Saved favorites are always just a click away in the favorites list. A highlighted star means you're on a favorite page. To remove a favorite, click the highlighted star. Your users can easily personalize their favorites lists by renaming favorites and rearranging items in the list.

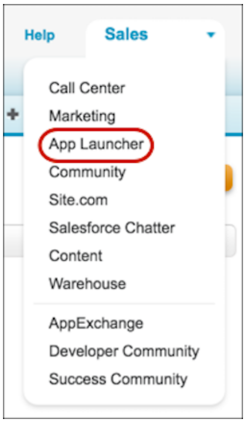
If an item in the navigation bar has a dropdown menu, it automatically includes the user's top favorites for the item.



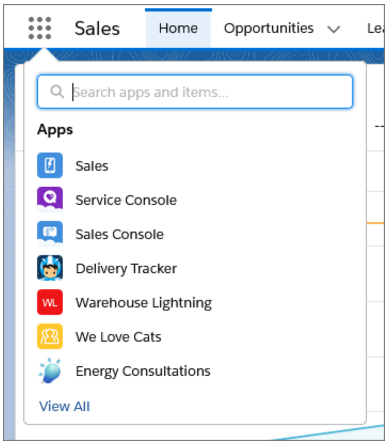
The App Launcher in Lightning Experience

In Lightning Experience, your users switch between apps through the App Launcher. They can browse the App Launcher to find available apps. Or they can search for an app by name from the Find an app or item box at the top of the App Launcher.

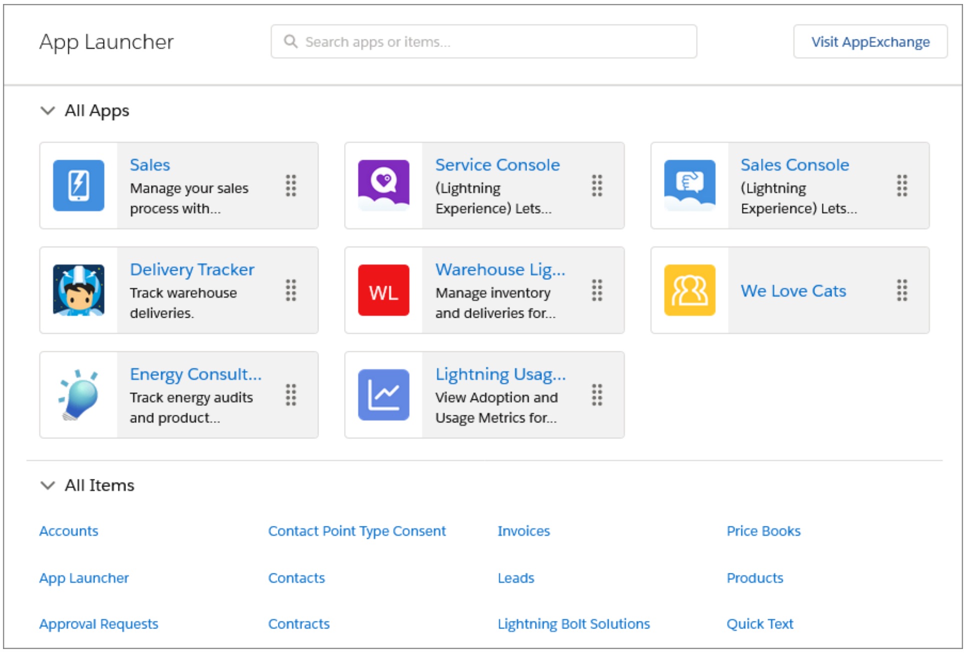
In Salesforce Classic, your users commonly switch between apps through the app menu or the App Launcher.



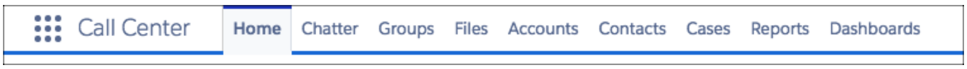
To get to the Lightning Experience App Launcher, click ⁙ from any page. The quick view menu contains your first 7 apps. To look for apps or items by name, use the Search apps and items box.



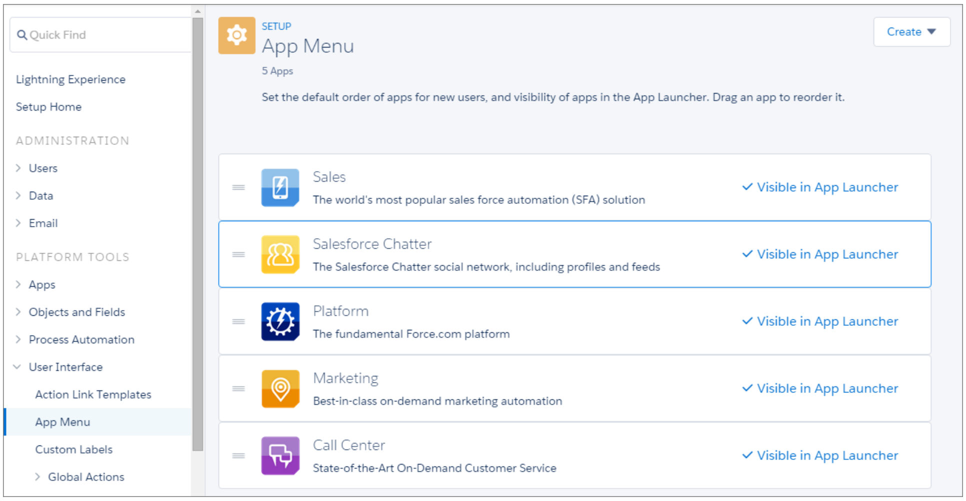
Or, click **View All** to see everything. Apps show up as large tiles under All Apps. Apps can include Salesforce standard apps, custom apps, and connected apps like G Suite. Other items, such as custom objects, tasks, events, and the feed, show up under All Items.



When you click in an app, you see the app name on the left side of the navigation bar. Items associated with the app appear to the right. Your users can create records and access recent records directly from the navigation bar for certain items like Accounts.



As a Salesforce admin, you can change which apps appear in the Lightning Experience App Launcher. You can also control the order in which the apps appear from the app menu.



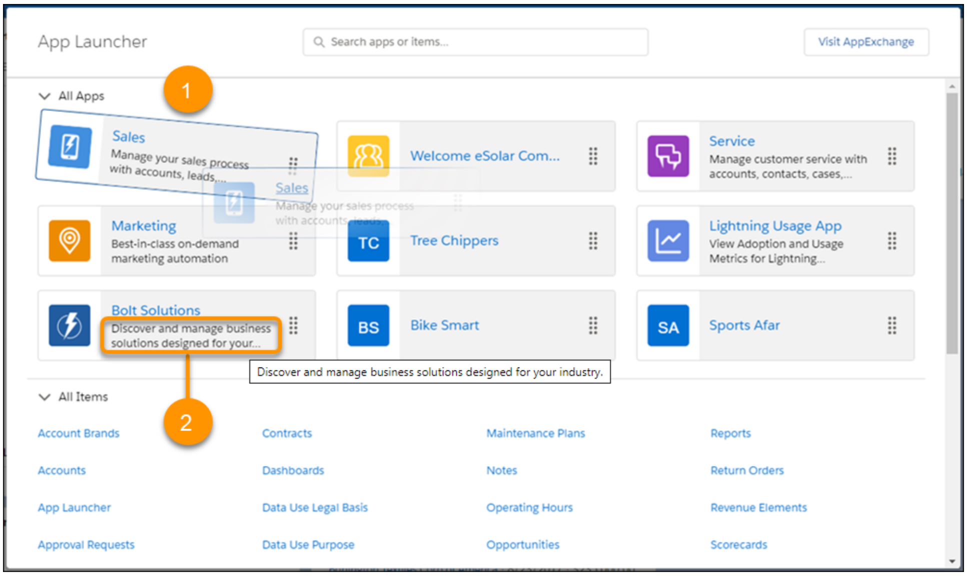
Set the default order of apps for new users, and visibility of apps in the App Launcher. Drag an app to reorder it.

Sales	The world's most popular sales force automation (SFA) solution	✓ Visible in App Launcher
Salesforce Chatter	The Salesforce Chatter social network, including profiles and feeds	✓ Visible in App Launcher
Platform	The fundamental Force.com platform	✓ Visible in App Launcher
Marketing	Best-in-class on-demand marketing automation	✓ Visible in App Launcher
Call Center	State-of-the-Art On-Demand Customer Service	✓ Visible in App Launcher

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 and app menu vary depending on each app's visibility settings and user permissions. Users see only the apps that they are authorized to see according to their profile or permission sets.

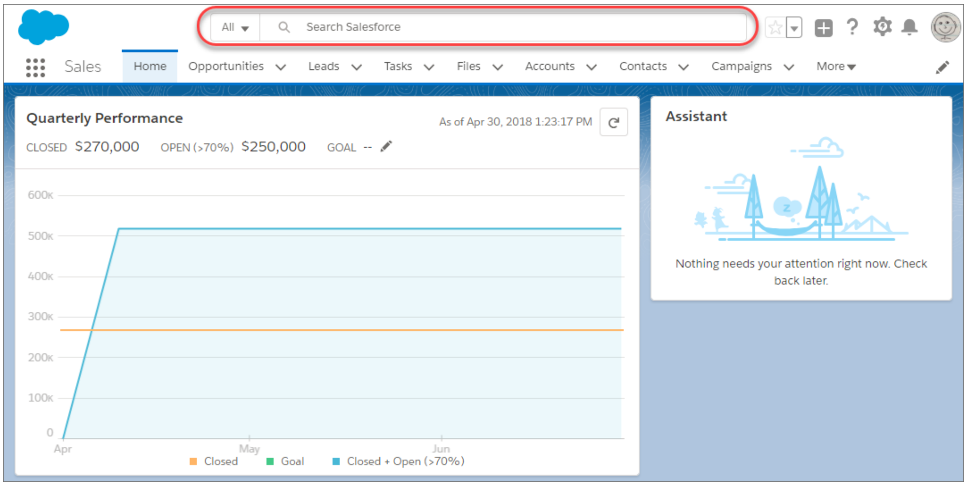
Users can drag tiles to sort their personal view of the App Launcher to their liking. Their sort order overrides the order that you set here.



Search for Records

You've got a ton of useful data in Salesforce. How do you get to what you need, when you need it? Let's face it, no one has time to browse anymore. It's all about search. So let us introduce you to the Lightning Experience global search box. It's at the top of every page, and it's the fastest way to bring what you need right to your fingertips.

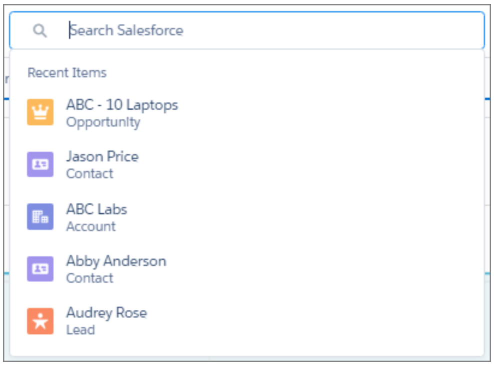
Think of search as your personal assistant—your very smart personal assistant. It anticipates your needs and helps you find what you're looking for from anywhere in the app.



Nothing needs your attention right now. Check back later.

Get Instant Results, Top Results, and More

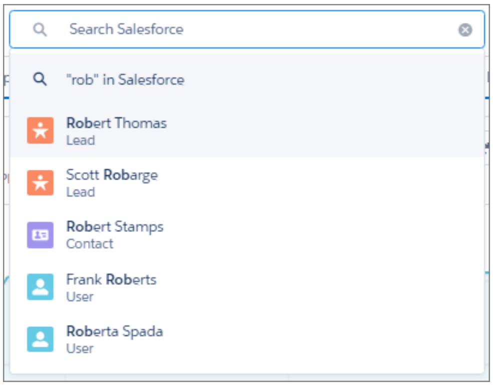
As soon as you click into the box, search starts working for you. A list of recent items provides quick links to records you've recently visited.



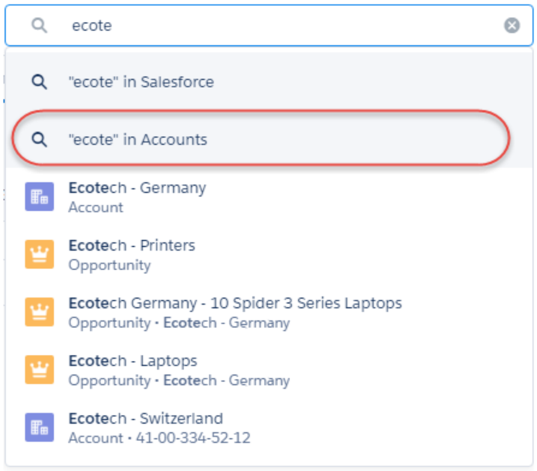
Recent Items

- ABC - 10 Laptops
  Opportunity
- Jason Price
  Contact
- ABC Labs
  Account
- Abby Anderson
  Contact
- Audrey Rose
  Lead

Start typing, and the list dynamically updates with matches from all searchable objects. If you see what you're looking for, select it to go right to the record.



Don’t see what you need in the dropdown list? Don’t worry—you’ve got options. To see results only for the object you’re on, select the second option in the instant results dropdown list.



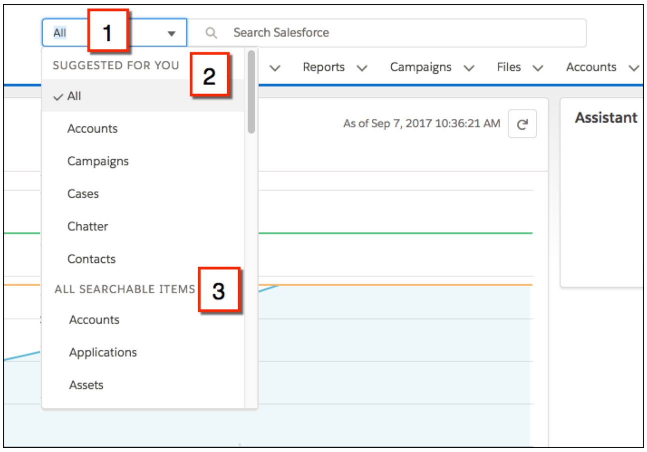
Alternatively, you can select the first option or press Enter to search across your entire organization. You’ll land on the Top Results page, which shows you the most relevant results from your most frequently used objects.

You also get a glimpse of how search helps you find what you need. Features like spelling correction and synonyms return matches that are similar to your search term. You might find one that’s just what you’re looking for.

Know which object you want to search? Start typing the name of the object next to the global search box. You can filter the dropdown list (1), or scroll to find what you're looking for. The objects you use most frequently are at the top (2), followed by all searchable objects, listed alphabetically (3).

To select a highlighted object, click it or press Enter. As you type your search term, the instant results are limited to the selected object. Press Enter to see full search results for only that object.

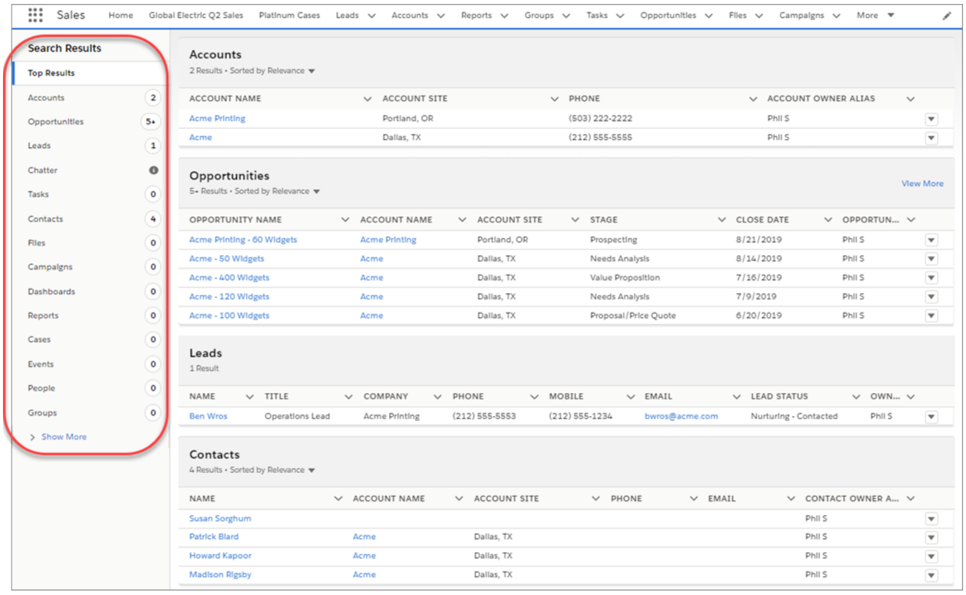
To remove the filter, change the object to **All**.



Refine Your Results

From the high-level overview of Top Results, it's easy to zoom in on what you need. See results for a specific object by selecting it on the left, under Search Results. You can see how many results were found for each object.

If you don't see an object that you need under Search Results, don't worry, it's close at hand. Select **Show More** to see all objects available to you, listed in alphabetical order.



Accounts	2
Opportunities	5+
Leads	1
Chatter	
Tasks	0
Contacts	4
Files	0
Campaigns	0
Dashboards	0
Reports	0
Cases	0
Events	0
People	0
Groups	0

Accounts

2 Results • Sorted by Relevance

ACCOUNT NAME	ACCOUNT SITE	PHONE	ACCOUNT OWNER ALIAS
Acme Printing	Portland, OR	(503) 222-2222	Phil S
Acme	Dallas, TX	(212) 555-5555	Phil S

Opportunities

5+ Results • Sorted by Relevance

OPPORTUNITY NAME	ACCOUNT NAME	ACCOUNT SITE	STAGE	CLOSE DATE	OPPORTUN...
Acme Printing - 60 Widgets	Acme Printing	Portland, OR	Prospecting	8/21/2019	Phil S
Acme - 50 Widgets	Acme	Dallas, TX	Needs Analysis	8/14/2019	Phil S
Acme - 400 Widgets	Acme	Dallas, TX	Value Proposition	7/16/2019	Phil S
Acme - 120 Widgets	Acme	Dallas, TX	Needs Analysis	7/9/2019	Phil S
Acme - 100 Widgets	Acme	Dallas, TX	Proposal/Price Quote	6/20/2019	Phil S

Leads

1 Result

NAME	TITLE	COMPANY	PHONE	MOBILE	EMAIL	LEAD STATUS	OWN...
Ben Wros	Operations Lead	Acme Printing	(212) 555-5553	(212) 555-1234	bwros@acme.com	Nurturing - Contacted	Phil S

Contacts

4 Results • Sorted by Relevance

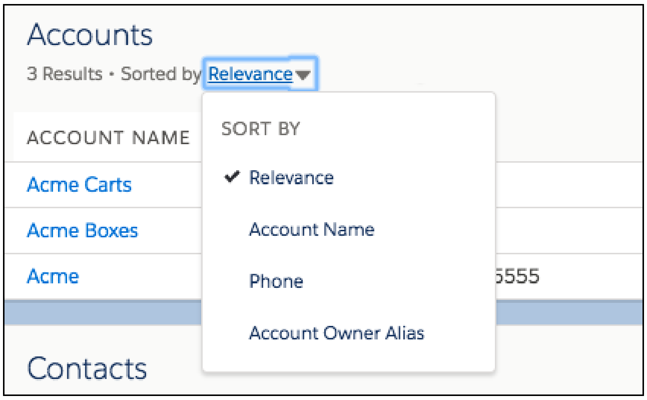
NAME	ACCOUNT NAME	ACCOUNT SITE	PHONE	EMAIL	CONTACT OWNER A...
Susan Sorghum					Phil S
Patrick Biard	Acme	Dallas, TX			Phil S
Howard Kapoor	Acme	Dallas, TX			Phil S
Madison Rigsby	Acme	Dallas, TX			Phil S

You can filter search results for accounts, cases, contacts, dashboards, files, Knowledge articles, leads, notes, opportunities, people, and tasks, and for custom objects. To see filtering options, click the object name in the Search Results sidebar.

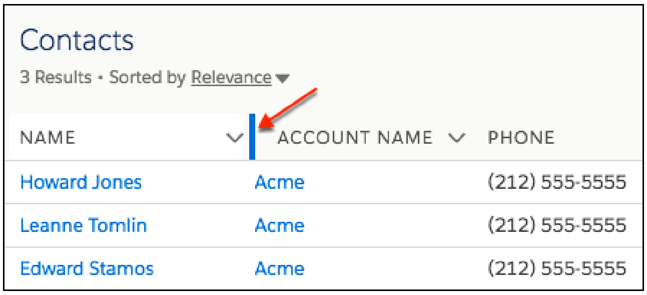
You can also sort most search results. When you first get to the search results page, you see that results are sorted by relevance. That sounds smart—and it is!—but what does it mean, exactly? Several things are considered, like:

- How unique is the search term?
- How often does the search term appear in a record?
- Do you own the record you're searching for?

You can also sort search results by clicking column headers or using the sort dropdown menu.



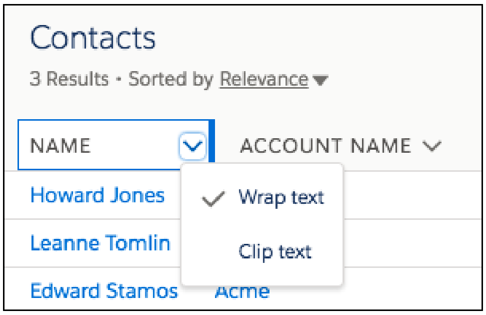
You can adjust column widths on the search results page by clicking and dragging the borders in the column header.


Contacts

3 Results • Sorted by Relevance

NAME	ACCOUNT NAME	PHONE
Howard Jones	Acme	(212) 555-5555
Leanne Tomlin	Acme	(212) 555-5555
Edward Stamos	Acme	(212) 555-5555

Text wrapping is on by default, but you can change this preference for a column by clicking the down arrow in the column header.



Can't Find What You're Searching For?

- If you get too many results, try using more—and more specific—search terms.
- Check whether the object or field is searchable.
- Make sure that you have access to the record. Search only returns results you have permission to view.
- If you recently created or updated the record, wait a few minutes for the record to be indexed. If you can't find your record after 15 minutes, contact your admin.

For more tips, see Search for Records in the Resources section.

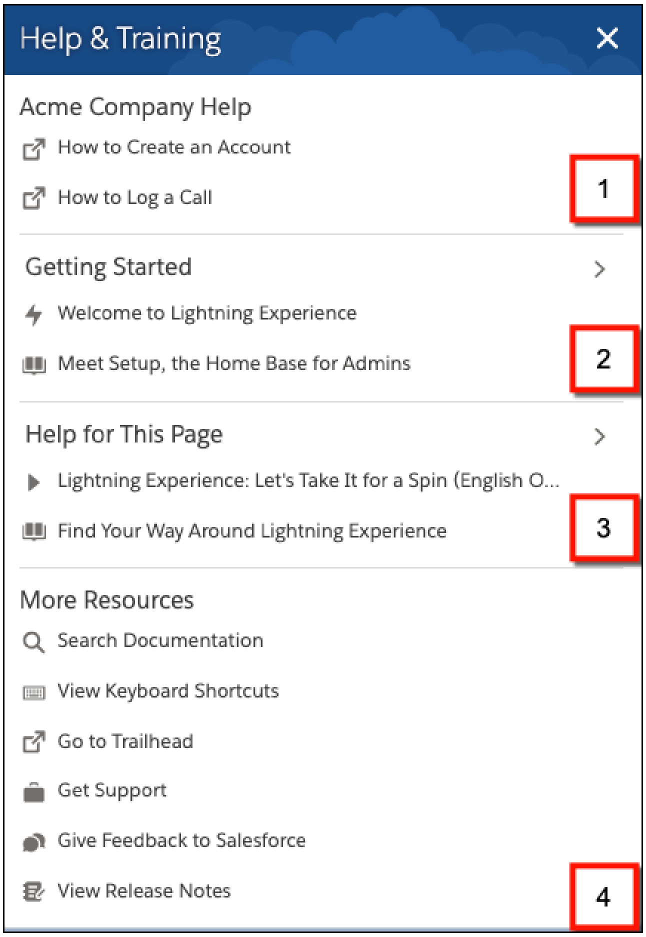
We think you'll agree you've found the perfect assistant—it has a broad perspective, yet directs you to the important things. It saves you so much time and effort that you'll probably wonder how you'd get your work done without it.

Lightning Experience Help Menu

In Lightning Experience, each page has a contextual Help Menu with links to resources related to the tasks on that page and resources for getting started. View help topics directly in the Help Menu, and dock them to keep them open while you work.

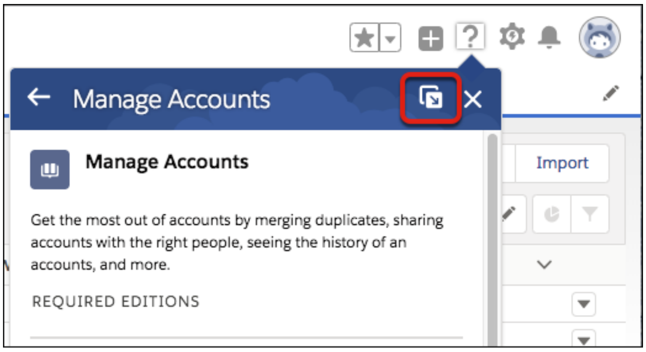
On object pages, the menu replaces the "Help for this Page" link that you're used to seeing in Salesforce Classic. The link still appears on many Setup pages, but we encourage you to use the Help Menu (? in the header) instead for more useful information.

If an admin added custom resources for working in an org, users see those resources first (1). When you're new to Lightning Experience or want some fresh tips, check out Getting Started (2). Watch helpful videos and read informative topics and Trailhead content on your journey to becoming a Lightning Experience pro. Admins and users see different sets of resources that are appropriate to their work in Salesforce. As you read, more suggested resources are displayed, so you're continually learning something new.



When appropriate, Salesforce suggests help topics, videos, Trailhead modules, and more under Help for This Page (3). If there's no suggested resources for the page, the section is hidden. For general tasks, such as searching documentation, giving feedback to Salesforce, or learning about keyboard shortcuts, look under More Resources (4). Admins see a link to view release notes.

Getting help doesn't mean leaving the app. When you click a help topic ( ) in the Help Menu, the topic opens in the Help menu. To keep the topic open while you work, click the docking icon.

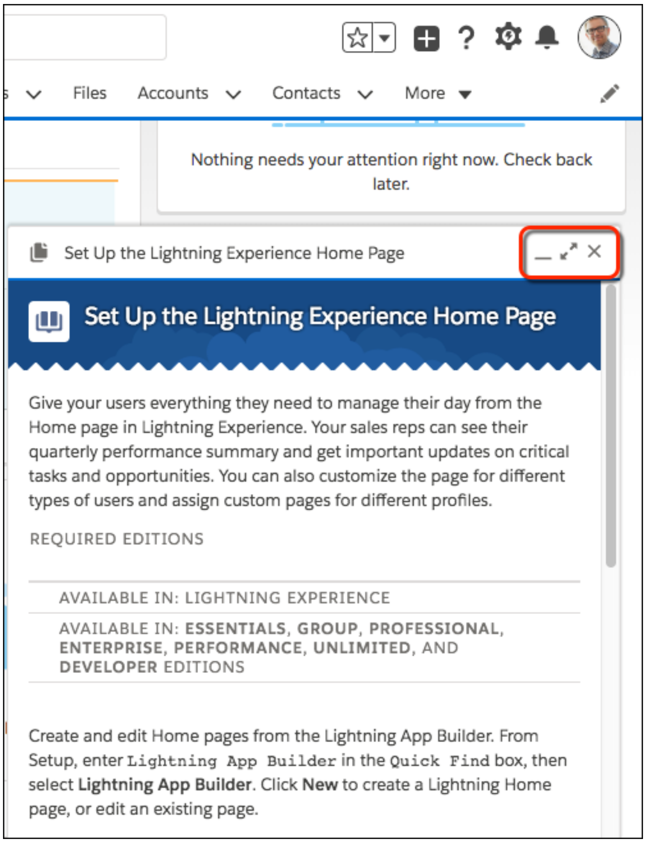


Manage Accounts

Get the most out of accounts by merging duplicates, sharing accounts with the right people, seeing the history of an accounts, and more.

REQUIRED EDITIONS

The topic stays available at the bottom of your screen. You can minimize it, expand it, or close it.



Nothing needs your attention right now. Check back later.

Set Up the Lightning Experience Home Page

Give your users everything they need to manage their day from the Home page in Lightning Experience. Your sales reps can see their quarterly performance summary and get important updates on critical tasks and opportunities. You can also customize the page for different types of users and assign custom pages for different profiles.

REQUIRED EDITIONS

AVAILABLE IN: LIGHTNING EXPERIENCE

AVAILABLE IN: **ESSENTIALS**, **GROUP**, **PROFESSIONAL**, **ENTERPRISE**, **PERFORMANCE**, **UNLIMITED**, AND **DEVELOPER** EDITIONS

Create and edit Home pages from the Lightning App Builder. From Setup, enter `Lightning App Builder` in the `Quick Find` box, then select **Lightning App Builder**. Click **New** to create a Lightning Home page, or edit an existing page.

For those new to Lightning Experience, be sure to check out **Getting Started** to get up to speed on the basics. Different resources are shown to users and admins. As you read the suggested resources, more topics are rotated to the top of the list, so you're continually learning something new.

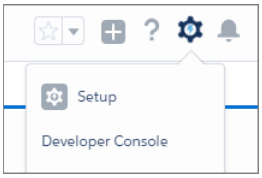
The Help Menu is no longer just for resources created by Salesforce. Admins can add a custom help section at the top of the Help Menu on every page in Lightning Experience. Guide users as they work in your org with links to your own URLs to websites, PDF files, videos, or Trailhead mixes. You get to name the section and pick the resources to display.

In case users still need more help, they can search through Salesforce documentation from within the Help Menu. Click **Search Documentation**, type in a search term, and see a list of results. Then, read a topic inside the Help Menu. Get your question answered and get back to the job at hand right away.

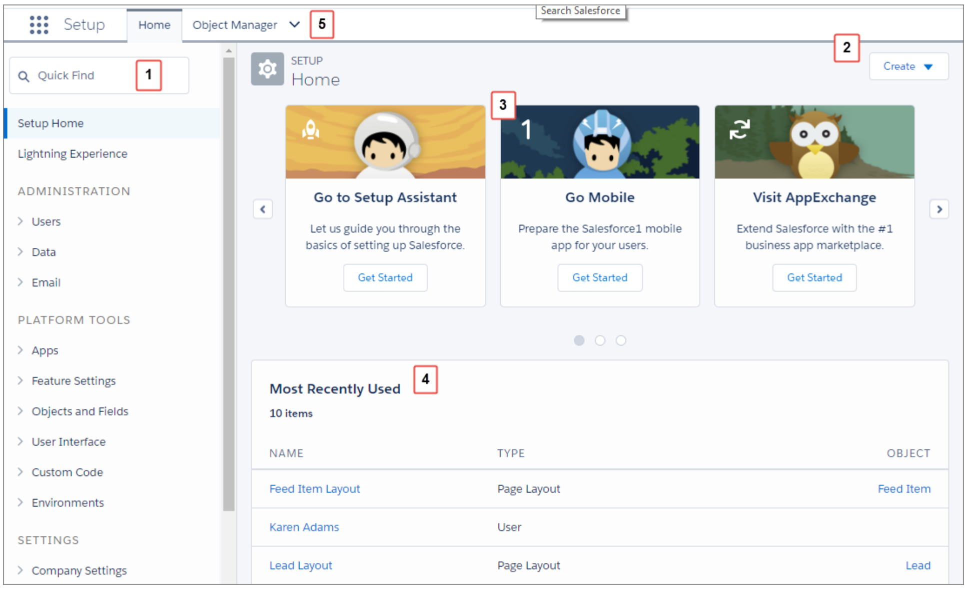
Meet the New and Improved Setup

A nip here? A tuck there? No, we've given Setup a whole new face!

You can navigate to Setup from the top of any page in Lightning Experience by clicking ⚙ > **Setup**.



The Setup tree has been completely reorganized and recategorized. In Salesforce Classic, the Setup tree had a lot of specific node categories, often with several nodes as nested subcategories. Sometimes there were many different ways to get to a destination. You might have mastered the click paths after a while, but to new users, this structure was often an overwhelming hurdle. The new Setup provides a streamlined interface for viewing and managing your administrative setup tasks.



NAME	TYPE	OBJECT
Feed Item Layout	Page Layout	Feed Item
Karen Adams	User	
Lead Layout	Page Layout	Lead

The new Setup includes these enhancements.

- The Quick Find (1) lets you quickly navigate to any node using a keyword. Quick Find is the best way to find what you're looking for if you know its name. Quick Find is your power tool for getting where you need to go!
- The Create menu (2) gives you quick access to common Setup creation functions—including users, custom objects, custom tabs, apps, email templates, and processes—without having to drill down through the Setup tree to get the page. You can get to the Create menu from any page in Setup.
- A carousel of quick-access tiles (3) gives you instant access to important setup tools and information, as well as the release notes. The Lightning Experience tile and Setup Salesforce tile help you enable your company for the new and improved UI, and mobile data access. There's also a link to download SalesforceA—which lets you do Salesforce administration from a mobile app—and a link to the System Status screen so you can view your org's performance and usage data.
- The Most Recently Used list (4) on the Setup Home page shows your most recently used records or customization features in Setup. You can quickly link back to what you were working on by clicking its name.
- The Object Manager (5) provides a one-stop shop for managing all objects in your organization, both standard and custom. We'll look at the object manager in more detail shortly.

Users can access their personal settings at the top of any page by clicking their profile image, then clicking **Settings**.

Administration, Platform Tools, and Settings, Oh My!

In the improved Setup, we've changed the five Setup tree sections from Administer, Build, Deploy, and Checkout to three sections: Administration, Platform Tools, and Settings. We completely reorganized all the child nodes to fit into these sections where appropriate, and added broader subcategories to make finding nodes easier, even if you don't know the exact name you're looking for.

Not only that, there are fewer child nodes to choose from, as many repetitive nodes in the tree have been removed. We reorganized the platform tools into a more process-oriented organizational structure rather than being feature-oriented. Now you can see pieces of the application life cycle broken up into subcategories: Apps, Objects and Fields, Process Automation, User Interface, Custom Code, and Environments.

At the bottom of the tree, in the Settings section, you can view company information or configure security.

Note: As you're getting familiar with the Setup area, it's important to keep in mind that navigating through Setup is not about memorizing click paths; it's about understanding what you're looking for in order to get to your destination. Depending on your users' profile and permissions, one user might see a different set of items in Setup than another. As a System Administrator, however, you see everything.

Where Did Some of the Nodes Go?

You might have noticed that some of the nodes are missing altogether. Only nodes that are related to customizing the new Lightning Experience are included in the Setup tree. For example, the Service Cloud related nodes are gone, as well as some of the Sales Cloud features. And, we moved all the object-related nodes to the Object Manager, which we'll look at shortly.

Don't worry! From a customization and development standpoint, all of the tools are still there. And as the other features become supported in Lightning Experience, you'll see them in Setup too.

Five Things You Shouldn't Miss in the Improved Setup

Lightning Experience Transition Assistant	• Your one stop for transitioning from Salesforce Classic to Lightning Experience • Step-by-step guidance on recommended activities and best practices
Create Menu	• On every page in Setup

	• Quick access to create common items
Object Manager	• All standard and custom objects now live in the Object Manager • All objects now have a standard detail page that stays visible while you drill into related lists • Infinite scroll on all objects' related lists
App Menu	Use it to: • Reorder apps in the App Launcher • Make apps visible or invisible in the App Launcher
View Release Notes	• Links to the most recent version of the release notes • Great point of reference for new and existing features

Limitations

Advanced Setup Search isn't available in Lightning Experience. However, you can work around that while in Setup by entering a term in global search and selecting the **in Setup** option in instant results. The search results page lists records that match your search term.

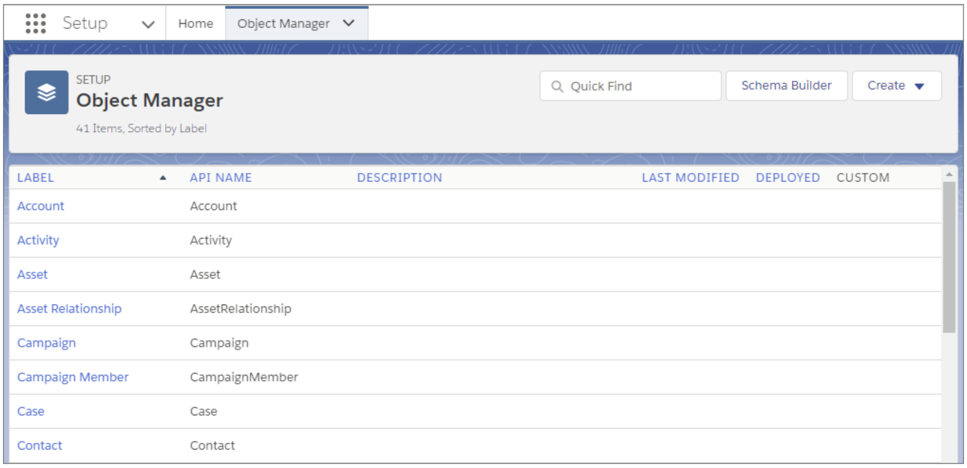
The Setup tree in Lightning Experience is limited to:

- Pages that support Lightning Experience features
- Administration pages that apply across your organization, such as user management, security, and company settings

Use Salesforce Classic to access administration pages for features that aren't in Lightning Experience.

Object Manager

The Object Manager is a one-stop shop for managing all objects in your organization, both standard and custom. Access all objects and their related functions—fields, validation rules, page layouts, and so on—from a single entry point.



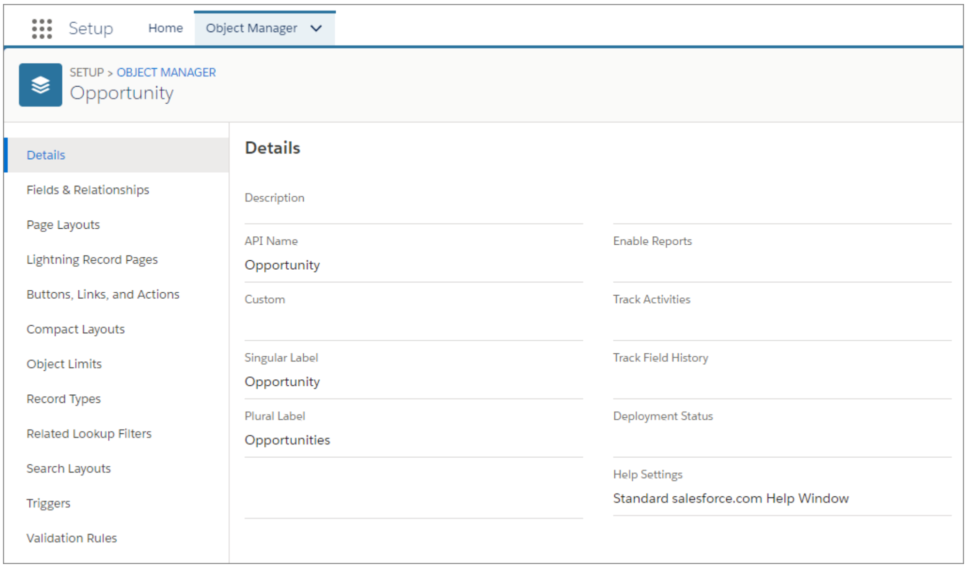
SETUP
Object Manager
41 Items, Sorted by Label

LABEL	API NAME	DESCRIPTION	LAST MODIFIED	DEPLOYED	CUSTOM
Account	Account				
Activity	Activity				
Asset	Asset				
Asset Relationship	AssetRelationship				
Campaign	Campaign				
Campaign Member	CampaignMember				
Case	Case				
Contact	Contact				

To access the Object Manager, from Setup, click **Object Manager**.

- To find an object, enter the first few characters of its label or name in the Quick Find box.
- To edit a custom object, click ▾ , then **Edit**.
- To view more details about an object or to access its related functions, click the object label.

From the object detail page, you can view the object details and access all related functions, such as fields, validation rules, and page layouts. To quickly jump to a function or control, use the links in the sidebar.



Details

Description	
API Name: Opportunity	Enable Reports
Custom	Track Activities
Singular Label: Opportunity	Track Field History
Plural Label: Opportunities	Deployment Status
	Help Settings: Standard salesforce.com Help Window

Limitations

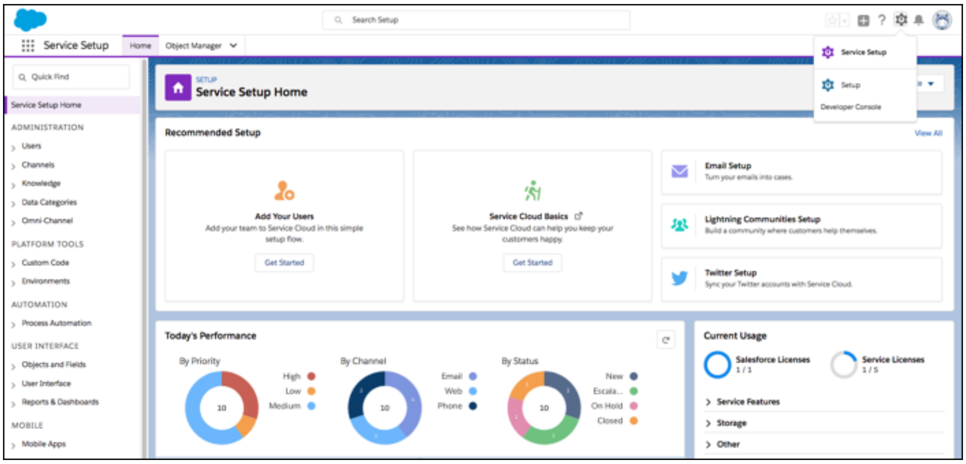
The Object Manager is limited to objects that support Lightning Experience features.

These object functions aren't listed in the Object Manager. You can access them from elsewhere in Setup.

- Case Comment Triggers
- Feed Comment Triggers
- Feed Item Triggers
- Feed Item Layouts
- Group Layouts
- Group Triggers
- Group Member Triggers
- Group Record Triggers
- Publisher Layouts
- Topic Triggers
- Topic Assignment Triggers

Service Setup Flows

Get your service team up and running quickly on a variety of support channels. Service Setup flows offer a guided step-by-step experience to walk you through setting up various support features like email support, integrating with Twitter and Facebook, and enabling a Knowledge base. Service Setup in Lightning Experience features an intuitive setup tree and a performance metrics dashboard too.



Add your team to Service Cloud in this simple setup flow.

See how Service Cloud can help you keep your customers happy.

Turn your emails into cases.

Build a community where customers help themselves.

Sync your Twitter accounts with Service Cloud.

The tiles across the top of the Service Setup Home page provide recommended setup flows, content, and one-step experiences that take the guesswork out of ramping up Service Cloud. The dashboard is customizable and shows at-a-glance metrics about your customer service team with daily performance or long-term trends. These insights help you assess and anticipate bottlenecks, and streamline calls to increase productivity.

CHAPTER 4 Opportunities, Leads, and Selling in Lightning Experience

In this chapter ...

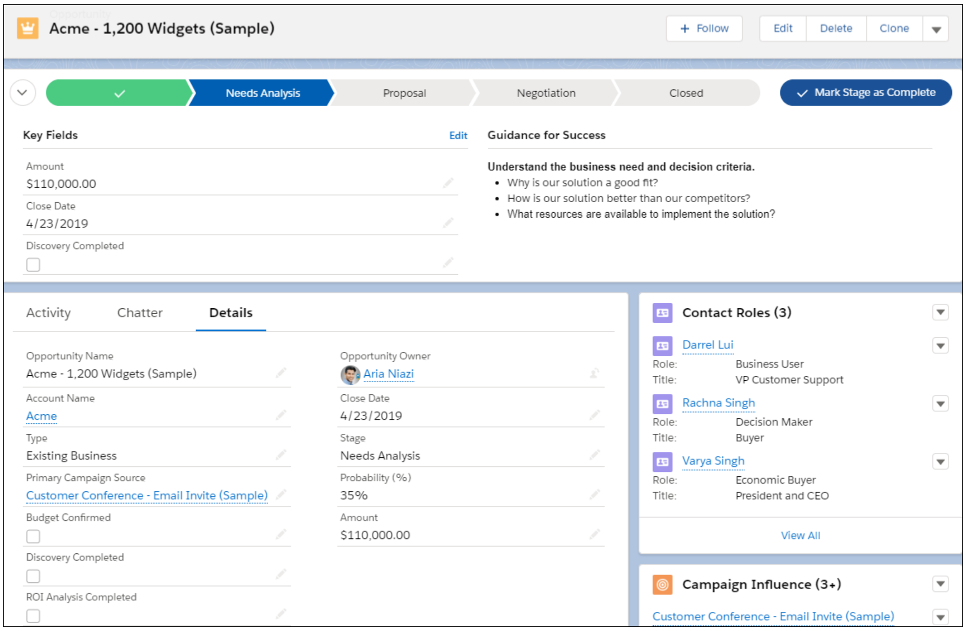
We've taken your sales process and put it into an action-first workspace, designed to help your sales reps work their deals faster and smarter. You can customize coaching scripts for each step in the sales process, create records quickly with fewer clicks, and ultimately close deals faster.

Explore the Opportunity Workspace

Opportunities have gotten a makeover! When you visit an opportunity record in Lightning Experience, you'll see some great enhancements that help your sales users get the most out of their opportunities.

Here are just a few things your users can do from the opportunity workspace:

- Create and update tasks and meetings, log calls, and send email
- View key information for a deal, like its key players, account, close date, amount, owner, and stage
- Update an opportunity's stage, close date, and amount
- View or add members to an opportunity team, including partner users, or remove a team
- View relevant, timely news about the account the opportunity is related to



Acme - 1,200 Widgets (Sample)

Key Fields

Amount
$110,000.00

Close Date
4/23/2019

Discovery Completed

Guidance for Success

Understand the business need and decision criteria.
- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

Opportunity Name: Acme - 1,200 Widgets (Sample)	Opportunity Owner: Aria Niazi
Account Name: Acme	Close Date: 4/23/2019
Type: Existing Business	Stage: Needs Analysis
Primary Campaign Source: Customer Conference - Email Invite (Sample)	Probability (%): 35%
Budget Confirmed:	Amount: $110,000.00
Discovery Completed:	
ROI Analysis Completed:	

Contact Roles (3)

Darrel Lui
Role: Business User
Title: VP Customer Support

Rachna Singh
Role: Decision Maker
Title: Buyer

Varya Singh
Role: Economic Buyer
Title: President and CEO

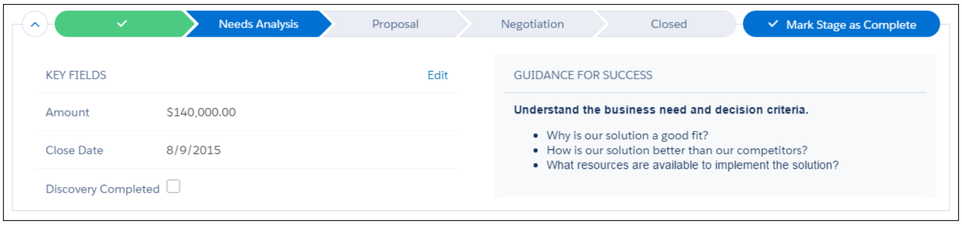
Campaign Influence (3+)

Customer Conference - Email Invite (Sample)

You don't have to do anything to get the opportunity workspace working for your users. However, as an admin, you can enhance your sales users' workflow by customizing the sales path and the activity timeline to their needs. And if your company uses a team selling approach, you can split your opportunity revenue by using opportunity splits in Lightning Experience.

Path

Paths guide your sales users through each stage of your company's sales process. Paths help users stay focused on important tasks so they can close deals or complete work quickly.

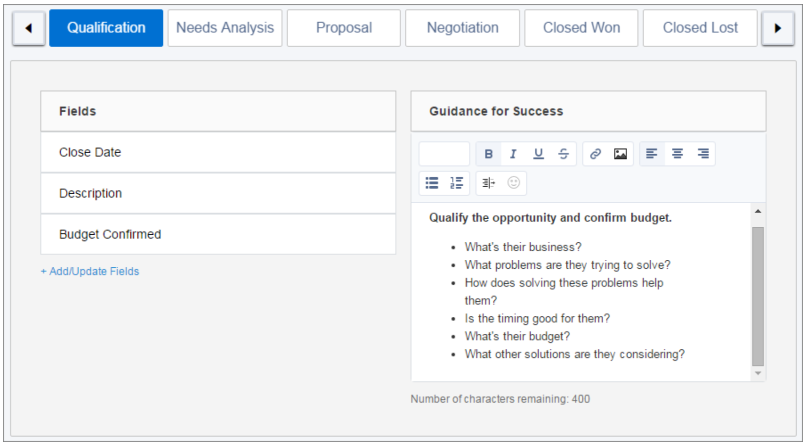


Amount	$140,000.00
Close Date	8/9/2015
Discovery Completed	☐

Understand the business need and decision criteria.

- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

From Setup, enter `Path Settings` in the `Quick Find` box, then select **Path Settings**. You can create a unique path for each record type for leads, opportunities, quotes, and custom objects.



Qualify the opportunity and confirm budget.

- What's their business?
- What problems are they trying to solve?
- How does solving these problems help them?
- Is the timing good for them?
- What's their budget?
- What other solutions are they considering?

Number of characters remaining: 400

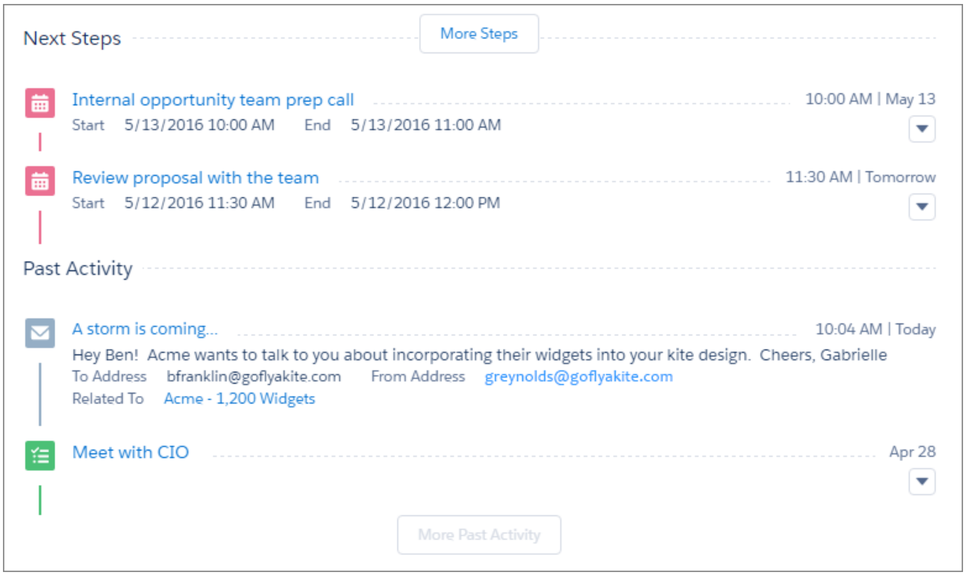
Path Best Practices

- Provide guidance for success content, like links to Chatter posts and videos, tips, or policy reminders—anything that can help sales reps get closer to sealing the deal.
- Keep your system performance optimal by creating sales paths that have 20 or fewer stages.
- Consider labeling sales paths for regions or industries, like "Steel Industry Sales Path."

- If you set up record types, you can have one path for each record type. For example, in the record type New Business, include more prospecting-related fields, but in the record type Existing Customer, include a field or stage for renewals.

Activity Timeline

With the activity timeline, your sales reps can keep a finger on the pulse of their deals. The timeline tracks meetings, tasks, calls, and emails. Reps can see what they've done and what they still have left to do for each opportunity, lead, account, and contact.



Internal opportunity team prep call — 10:00 AM | May 13
Start 5/13/2016 10:00 AM End 5/13/2016 11:00 AM

Review proposal with the team — 11:30 AM | Tomorrow
Start 5/12/2016 11:30 AM End 5/12/2016 12:00 PM

A storm is coming... — 10:04 AM | Today
Hey Ben! Acme wants to talk to you about incorporating their widgets into your kite design. Cheers, Gabrielle
To Address bfranklin@goflyakite.com From Address greynolds@goflyakite.com
Related To Acme - 1,200 Widgets

Meet with CIO — Apr 28

Activity Timeline Considerations

When working with the activity timeline, keep the following in mind.

The configuration of page layouts and record types affects the tabs in the activity composer

Don't see the tabs for calls, tasks, events, or emails in the activity composer? Adjust your page layouts, record types, and user permissions. See Add Send an Email, Log a Call, New Event, and New Task Buttons to the Activity Composer.

You can customize the display and order of fields in the activity timeline

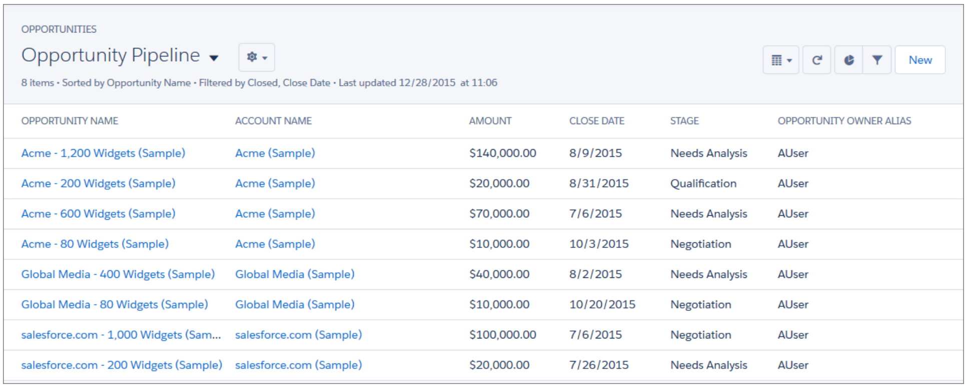
In the activity timeline, you can customize the display and order of fields when an activity is expanded for events, tasks, and logged calls. Do this customization using event and task compact layouts. Even if you remove certain fields from a compact layout, they remain in the activity timeline because they contain essential activity information. For example, suppose that you remove the due date, the date and time, or the task status fields from a compact layout. The event start date and time, the task checkbox, the task due date, and the call logging date still appear on activities in the timeline. The description field for events and the comments field for tasks also always appear in the timeline, although they aren't available in the compact layout. The remaining fields visible in the timeline reflect the fields you include in the compact layout.

The activity timeline icons aren't customizable

The icons for activity types (events, tasks, calls, and email) in the timeline aren't customizable.

Opportunity Home

The Opportunity home page in Lightning Experience looks a lot like other object home pages at first glance. You might not notice it at first, but an awesome feature is waiting to be discovered there.

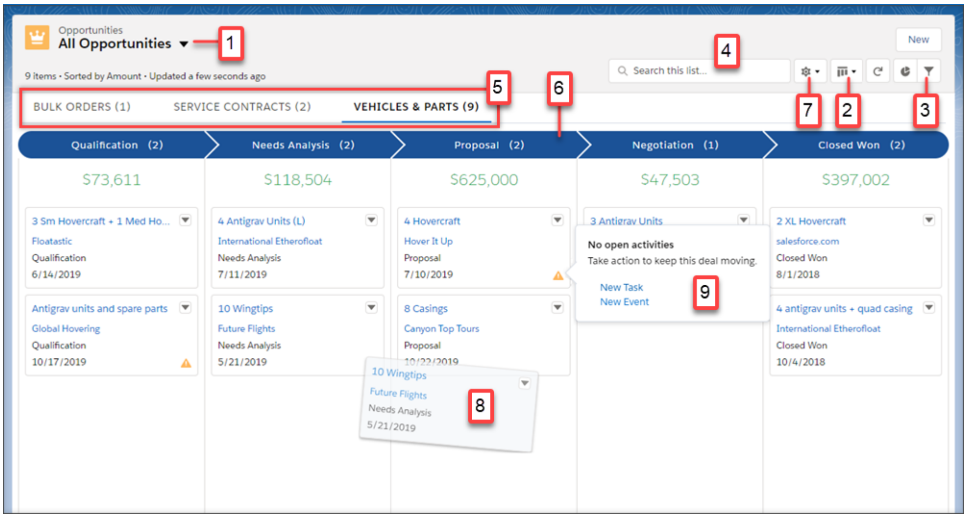


8 items • Sorted by Opportunity Name • Filtered by Closed, Close Date • Last updated 12/28/2015 at 11:06

OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORTUNITY OWNER ALIAS
Acme - 1,200 Widgets (Sample)	Acme (Sample)	$140,000.00	8/9/2015	Needs Analysis	AUser
Acme - 200 Widgets (Sample)	Acme (Sample)	$20,000.00	8/31/2015	Qualification	AUser
Acme - 600 Widgets (Sample)	Acme (Sample)	$70,000.00	7/6/2015	Needs Analysis	AUser
Acme - 80 Widgets (Sample)	Acme (Sample)	$10,000.00	10/3/2015	Negotiation	AUser
Global Media - 400 Widgets (Sample)	Global Media (Sample)	$40,000.00	8/2/2015	Needs Analysis	AUser
Global Media - 80 Widgets (Sample)	Global Media (Sample)	$10,000.00	10/20/2015	Negotiation	AUser
salesforce.com - 1,000 Widgets (Sam...	salesforce.com (Sample)	$100,000.00	7/6/2015	Negotiation	AUser
salesforce.com - 200 Widgets (Sample)	salesforce.com (Sample)	$20,000.00	7/26/2015	Needs Analysis	AUser

Kanban View

The Kanban view for Opportunities is a visual representation of all of a sales rep's deals, organized by each stage in the pipeline. You can get to the Kanban view by selecting **Kanban** from the Displays menu on all list views except Recently Viewed.



$73,611
3 Sm Hovercraft + 1 Med Ho...
Floatastic
Qualification
6/14/2019
Antigrav units and spare parts
Global Hovering
Qualification
10/17/2019

$118,504
4 Antigrav Units (L)
International Etherofloat
Needs Analysis
7/11/2019
10 Wingtips
Future Flights
Needs Analysis
5/21/2019

$625,000
4 Hovercraft
Hover It Up
Proposal
7/10/2019
8 Casings
Canyon Top Tours
Proposal
10/22/2019

$47,503
3 Antigrav Units
No open activities
Take action to keep this deal moving.
New Task
New Event

$397,002
2 XL Hovercraft
salesforce.com
Closed Won
8/1/2018
4 antigrav units + quad casing
International Etherofloat
Closed Won
10/4/2018

10 Wingtips
Future Flights
Needs Analysis
5/21/2019

The records in the Kanban view are based on the selected list view (1). Reps can't view the Kanban for Recently Viewed list views. Easily toggle between the list view grid view and the Kanban view (2). Filter records to select a single record type or view a particular subset of records (3). Records are separated based on record type (4). Records are grouped into columns (5) based on the grouping field selected. Quickly move a record to a different stage by dragging the card (6). For opportunities, alerts tell how to keep a deal on track, for example, create a task or event (7). Quickly move a record to a different column by dragging the card (8). For opportunities, alerts tell how to keep a deal on track, for example, create a task or event (9).

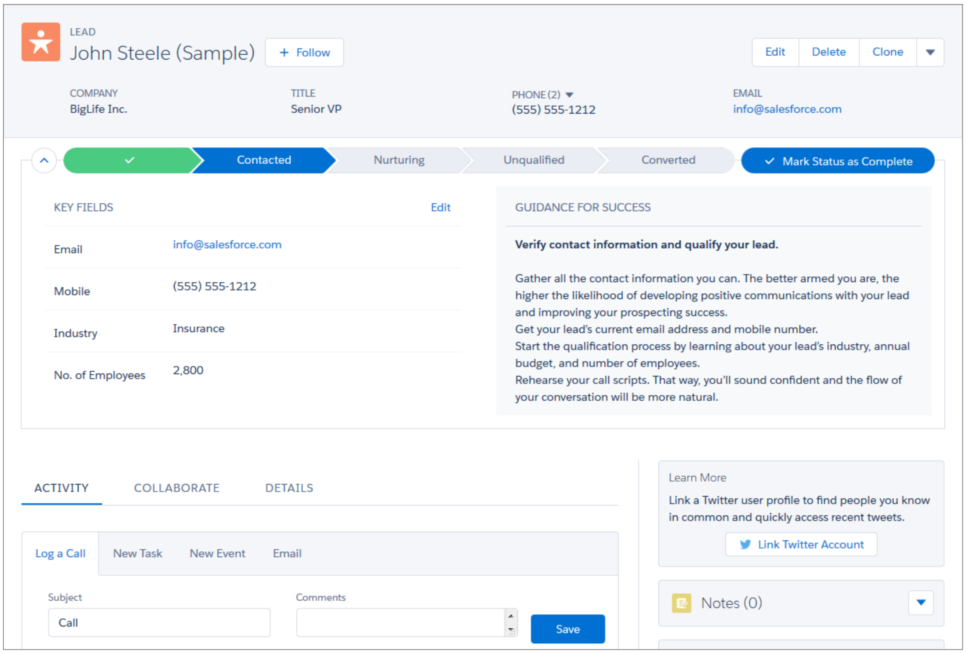
Here, your users can manage their opportunities through all phases of the pipeline, dragging and dropping opportunities from one column to another. A yellow triangle on an opportunity card can indicate three types of alerts: overdue tasks, no open activities, or no activity for 30 days. Users can click the triangle to create tasks and events right from the card. Items on the board vary based on which list view is open.

Tip: The Kanban view isn't just for opportunities--you can access a Kanban view for almost any list view.

Explore the Lead Workspace

Qualifying and converting leads is easy for your sales reps. Lightning Experience includes a lead workspace—command central where your reps track, update, and convert leads to contacts.

Just like with opportunities, the workspace for leads includes Path. You set up paths to include specific fields and guidance for success in each stage of the lead qualification process.



LEAD
John Steele (Sample)

COMPANY	TITLE	PHONE (2)	EMAIL
BigLife Inc.	Senior VP	(555) 555-1212	info@salesforce.com

KEY FIELDS

Email	info@salesforce.com
Mobile	(555) 555-1212
Industry	Insurance
No. of Employees	2,800

GUIDANCE FOR SUCCESS

Verify contact information and qualify your lead.

Gather all the contact information you can. The better armed you are, the higher the likelihood of developing positive communications with your lead and improving your prospecting success.
Get your lead's current email address and mobile number.
Start the qualification process by learning about your lead's industry, annual budget, and number of employees.
Rehearse your call scripts. That way, you'll sound confident and the flow of your conversation will be more natural.

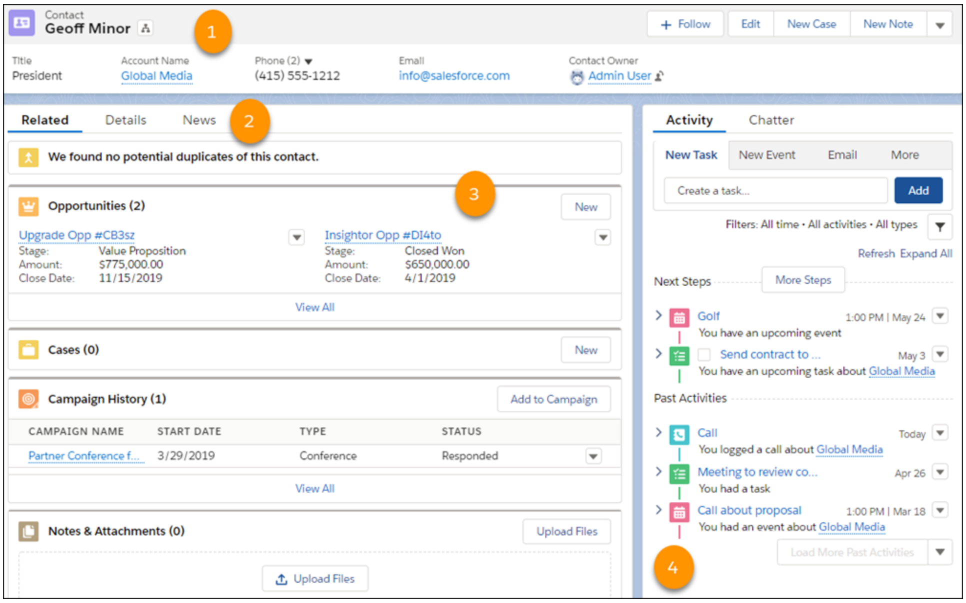
Learn More
Link a Twitter user profile to find people you know in common and quickly access recent tweets.

Subject: Call
Comments:

To convert a lead, your reps click the **Converted** stage in the path. Then, they either select an account or create one on the spot. Reps can also create an opportunity.

Accounts and Contacts in Lightning Experience

In Lightning Experience, we optimized business and person accounts and contacts for quick reference, so it's easy to find information and gather insights at a glance.



Contact
Geoff Minor

Title	Account Name	Phone (2)	Email	Contact Owner
President	Global Media	(415) 555-1212	info@salesforce.com	Admin User

We found no potential duplicates of this contact.

Opportunities (2)

Upgrade Opp #CB3sz
Stage: Value Proposition
Amount: $775,000.00
Close Date: 11/15/2019

Insightor Opp #DI4to
Stage: Closed Won
Amount: $650,000.00
Close Date: 4/1/2019

Cases (0)

Campaign History (1)

CAMPAIGN NAME	START DATE	TYPE	STATUS
Partner Conference f...	3/29/2019	Conference	Responded

Notes & Attachments (0)

Create a task...

Filters: All time • All activities • All types

Next Steps

Golf 1:00 PM \| May 24
You have an upcoming event

Send contract to ... May 3
You have an upcoming task about Global Media

Past Activities

Call Today
You logged a call about Global Media

Meeting to review co... Apr 26
You had a task

Call about proposal 1:00 PM \| Mar 18
You had an event about Global Media

1. Highlight panel: See important information right at the top of the record.
2. News and Twitter integration: Stay informed about the latest news that affects your customers and stay connected through social media.
3. Optimized template: Easily refer to related records and other information at a glance.
4. Activity timeline: View emails, tasks, and events, grouped by your next steps and past activity.

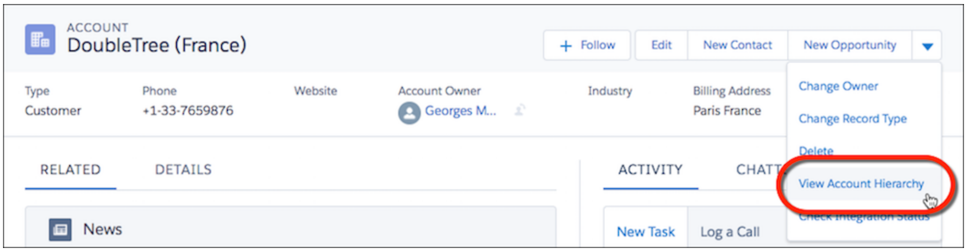
As with opportunity teams, your users can view or add members to an account team, including partner users, or remove a team.

There are other goodies too, like visualizing account relationships in an account hierarchy, seeing who reports to whom in a contact hierarchy, and relating single contacts to multiple accounts without duplicating records. Let's take a look.

Account and Contact Hierarchies

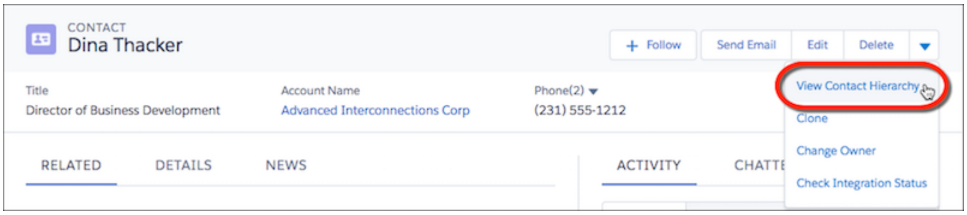
As in Salesforce Classic, an account hierarchy gives sales reps a bird's-eye view of the relationships between a parent account and its subsidiary accounts. A contact hierarchy lets sales reps generate an org chart for an account.

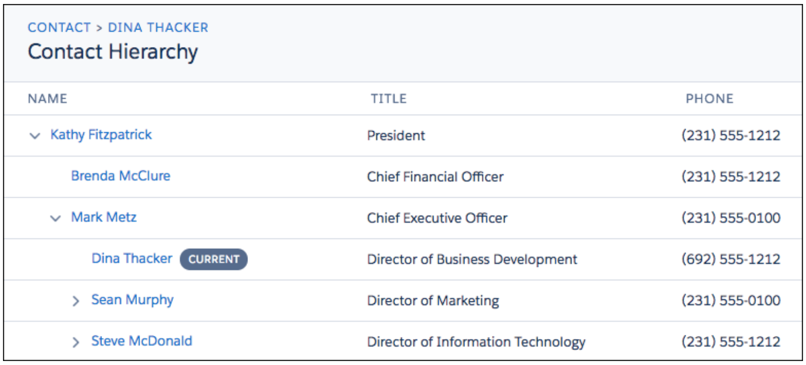
In Lightning Experience, view an account hierarchy using the Actions dropdown menu.



Users expand or collapse parts of a hierarchy as they navigate through it. They can view up to 2,000 accounts from each point where they enter a hierarchy. Account hierarchies are derived from the Parent Account field on accounts.

In Lightning Experience, view the contact hierarchy from the Actions dropdown menu on the contact record page.

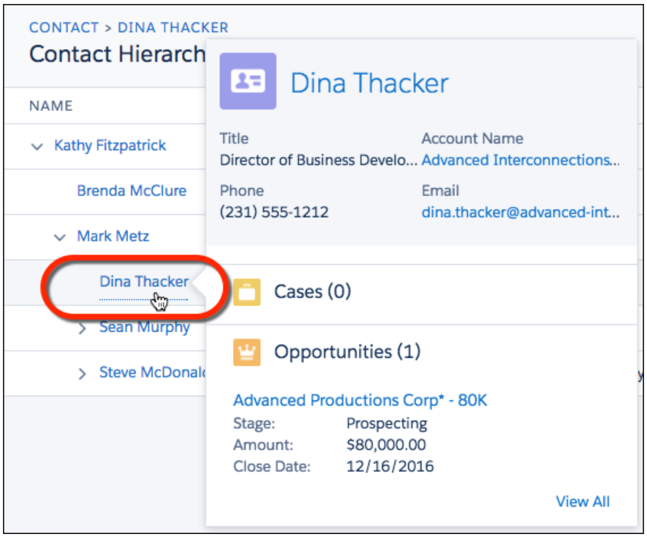



CONTACT > DINA THACKER

Contact Hierarchy

NAME	TITLE	PHONE
⌄ Kathy Fitzpatrick	President	(231) 555-1212
Brenda McClure	Chief Financial Officer	(231) 555-1212
⌄ Mark Metz	Chief Executive Officer	(231) 555-0100
Dina Thacker CURRENT	Director of Business Development	(692) 555-1212
› Sean Murphy	Director of Marketing	(231) 555-0100
› Steve McDonald	Director of Information Technology	(231) 555-1212

Hovering over a contact's name shows more details.



CONTACT > DINA THACKER

Contact Hierarch

NAME
Kathy Fitzpatrick
Brenda McClure
Mark Metz
Dina Thacker
Sean Murphy
Steve McDonal

Dina Thacker

Title	Account Name
Director of Business Develo...	Advanced Interconnections...
Phone	**Email**
(231) 555-1212	dina.thacker@advanced-int...

Cases (0)

Opportunities (1)

Advanced Productions Corp* - 80K

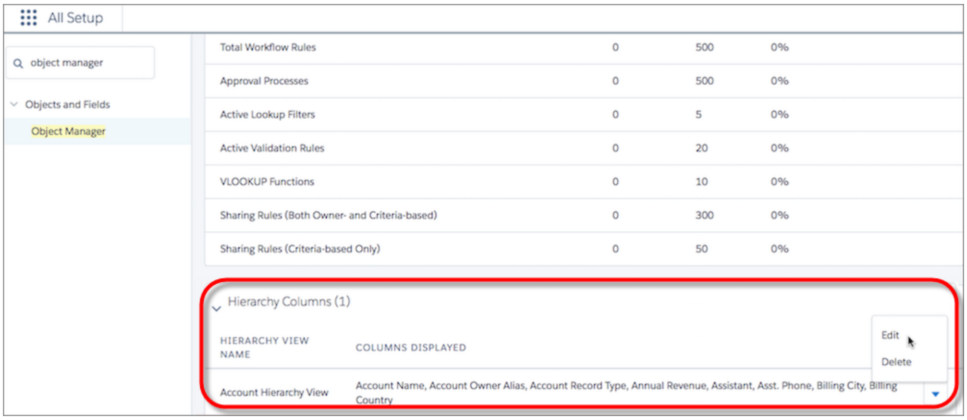
Stage:	Prospecting
Amount:	$80,000.00
Close Date:	12/16/2016

View All

Contact hierarchies are derived from the Reports To field on contact records.

There's a bonus to hierarchies in Lightning Experience—you can customize hierarchy columns to show the information that's most useful to your sales reps.

To customize the account hierarchy columns, from **Setup** in Lightning Experience, click the **Object Manager** tab. In **Account**, click **Hierarchy Columns**, and then edit the columns. This customization is independent of the standard Recently Viewed Accounts list view.



Total Workflow Rules	0	500	0%
Approval Processes	0	500	0%
Active Lookup Filters	0	5	0%
Active Validation Rules	0	20	0%
VLOOKUP Functions	0	10	0%
Sharing Rules (Both Owner- and Criteria-based)	0	300	0%
Sharing Rules (Criteria-based Only)	0	50	0%

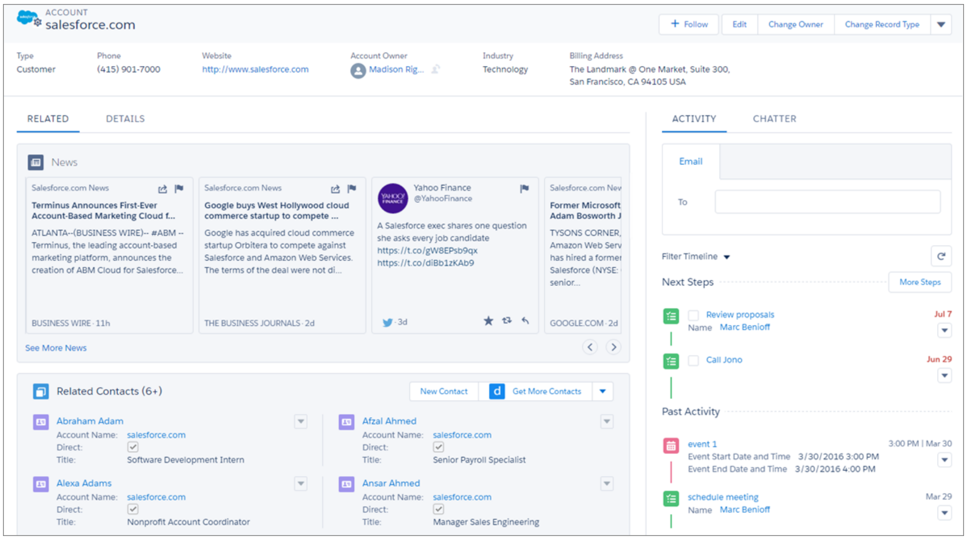
Hierarchy Columns (1)

HIERARCHY VIEW NAME	COLUMNS DISPLAYED
Account Hierarchy View	Account Name, Account Owner Alias, Account Record Type, Annual Revenue, Assistant, Asst. Phone, Billing City, Billing Country

Likewise, to customize the contact hierarchy columns, in Contact, click **Hierarchy Columns**, and then edit the columns.

Contacts to Multiple Accounts

The Contacts to Multiple Accounts feature lets you relate a single contact to multiple accounts so that you can easily track the relationships between people and businesses—without creating duplicate records. When the feature is set up, account records include the Related Contacts related list and contact records include the Related Accounts related list.



ACCOUNT
salesforce.com

+ Follow | Edit | Change Owner | Change Record Type

Type	Phone	Website	Account Owner	Industry	Billing Address
Customer	(415) 901-7000	http://www.salesforce.com	Madison Rig...	Technology	The Landmark @ One Market, Suite 300, San Francisco, CA 94105 USA

RELATED DETAILS

News

Salesforce.com News
Terminus Announces First-Ever Account-Based Marketing Cloud f...
ATLANTA--(BUSINESS WIRE)-- #ABM -- Terminus, the leading account-based marketing platform, announces the creation of ABM Cloud for Salesforce...
BUSINESS WIRE · 11h

Salesforce.com News
Google buys West Hollywood cloud commerce startup to compete ...
Google has acquired cloud commerce startup Orbitera to compete against Salesforce and Amazon Web Services. The terms of the deal were not di...
THE BUSINESS JOURNALS · 2d

Yahoo Finance
@YahooFinance
A Salesforce exec shares one question she asks every job candidate https://t.co/gW8EPsb9qx https://t.co/dIBb1zKAb9
· 3d

Salesforce.com New
Former Microsoft Adam Bosworth J
TYSONS CORNER, Amazon Web Serv has hired a former Salesforce (NYSE: senior...
GOOGLE.COM · 2d

See More News

Related Contacts (6+)

New Contact | Get More Contacts

Abraham Adam
Account Name: salesforce.com
Direct: ✓
Title: Software Development Intern

Afzal Ahmed
Account Name: salesforce.com
Direct: ✓
Title: Senior Payroll Specialist

Alexa Adams
Account Name: salesforce.com
Direct: ✓
Title: Nonprofit Account Coordinator

Ansar Ahmed
Account Name: salesforce.com
Direct: ✓
Title: Manager Sales Engineering

ACTIVITY CHATTER

Email
To

Filter Timeline

Next Steps — More Steps

Review proposals — Jul 7
Name Marc Benioff

Call Jono — Jun 29

Past Activity

event 1 — 3:00 PM | Mar 30
Event Start Date and Time 3/30/2016 3:00 PM
Event End Date and Time 3/30/2016 4:00 PM

schedule meeting — Mar 29
Name Marc Benioff

CHAPTER 5 Service Cloud in Lightning Experience

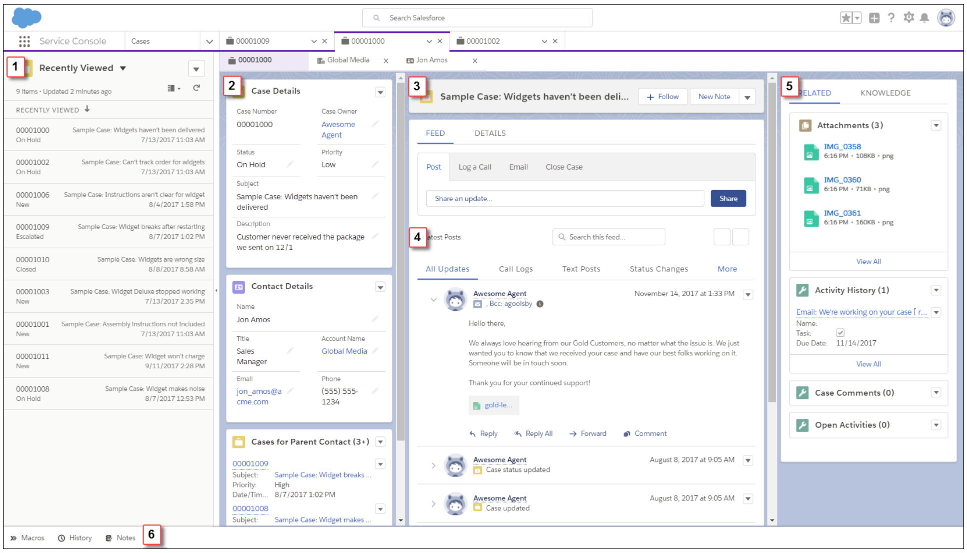
In this chapter ...

Wondering which Service Cloud features are available in Lightning Experience? Well, wonder no longer! Let's talk Service Cloud.

The Fast-As-Lightning Service Console

The Lightning Service Console is a standard Salesforce Lightning console app that's designed to meet all your service team's needs. Plus you can customize the console to make it your own.

Here's what you get in the standard Service Console app.



9 Items • Updated 2 minutes ago

RECENTLY VIEWED

00001000	On Hold	Sample Case: Widgets haven't been delivered 7/13/2017 11:03 AM
00001002	On Hold	Sample Case: Can't track order for widgets 7/13/2017 11:03 AM
00001006	New	Sample Case: Instructions aren't clear for widget 8/4/2017 1:58 PM
00001009	Escalated	Sample Case: Widget breaks after restarting 8/7/2017 1:02 PM
00001010	Closed	Sample Case: Widgets are wrong size 8/8/2017 8:58 AM
00001003	New	Sample Case: Widget Deluxe stopped working 7/13/2017 2:35 PM
00001001	New	Sample Case: Assembly Instructions not Included 7/13/2017 11:03 AM
00001011	New	Sample Case: Widget won't charge 9/11/2017 2:28 PM
00001008	On Hold	Sample Case: Widget makes noise 8/7/2017 12:53 PM

Case Details

Case Number: 00001000
Case Owner: Awesome Agent
Status: On Hold
Priority: Low
Subject: Sample Case: Widgets haven't been delivered
Description: Customer never received the package we sent on 12/1

Contact Details

Name: Jon Amos
Title: Sales Manager
Account Name: Global Media
Email: jon_amos@acme.com
Phone: (555) 555-1234

Cases for Parent Contact (3+)

00001009
Subject: Sample Case: Widget breaks ...
Priority: High
Date/Tim... 8/7/2017 1:02 PM
00001008
Subject: Sample Case: Widget makes ...

Sample Case: Widgets haven't been deli...

Awesome Agent , Bcc: agoolsby — November 14, 2017 at 1:33 PM

Hello there,

We always love hearing from our Gold Customers, no matter what the issue is. We just wanted you to know that we received your case and have our best folks working on it. Someone will be in touch soon.

Thank you for your continued support!

gold-le...

Awesome Agent — Case status updated — August 8, 2017 at 9:05 AM

Awesome Agent — Case updated — August 8, 2017 at 9:05 AM

Attachments (3)

IMG_0358 6:16 PM • 108KB • png
IMG_0360 6:16 PM • 71KB • png
IMG_0361 6:16 PM • 160KB • png

Activity History (1)

Email: We're working on your case [r...
Name:
Task: ✓
Due Date: 11/14/2017

Case Comments (0)

Open Activities (0)

Here's how the standard Service Console app helps you out.

Manage cases faster

Split view shows a list view at the same time as workspace tabs and subtabs, letting your support agents manage multiple cases on a single screen (1).

See it all at once

A preconfigured, three-column layout puts all the important information on the same page, and minimizes clicks and scrolling.

- In the first column, case details, contact details, and related cases for the parent account are displayed using Related Record and Related List Single components (2).
- The second column includes a highlights panel that's front and center (3). The compact case feed lets you see more of your case updates (4).
- The third column includes related lists and Knowledge articles relevant to your case (5).

Keep up on all the details

The preconfigured utility bar provides fast access to History and Notes (6). You can also customize the utility bar to include other tools like Macros, Omni-Channel, and the Open CTI Softphone.

Note: For users to see and use the Notes utility, Notes must be enabled in your org. For users to see and use articles in the Knowledge component, Lightning Knowledge must be enabled in your org.

Best Practice Makes Perfect

As awesome as the default Lightning Service Console is, with a little polish, you can tailor it to meet your specific service needs.

We recommend that you:

- Update the name, description, and branding for the app
- Add other items, such as quick text, macros, products, and contracts (the Service Console app includes cases, contacts, accounts, reports, dashboards, chatter, and Home by default)
- Add other utilities, such as an Open CTI softphone, Omni-Channel, Macros, and Community 360
- Assign the app to the appropriate user profiles, like your service agent profiles

Note: The Service Console app is automatically assigned to all standard and custom user profiles. You can modify these assignments to make sure that only the right people have access to the app.

The only thing you can't change is the app's Developer Name.

Be a Knowledge Sponge

So what's so great about Lightning Knowledge? Everything you need is all in one place! It provides a high-powered yet streamlined way to manage your knowledge base. You get the benefits of standard objects that work just like other objects in Salesforce. Lightning Knowledge has changed the way Knowledge works in Salesforce. Standard record types replace article types, and the Knowledge component for Lightning Service Console replaces Knowledge One for the Service Console in Salesforce Classic. These changes make it much easier for you to set things up.

Your support agents can search, view, author, and manage articles from a single Knowledge home page in Lightning Experience. Plus, they can do the following authoring actions without leaving Knowledge home.

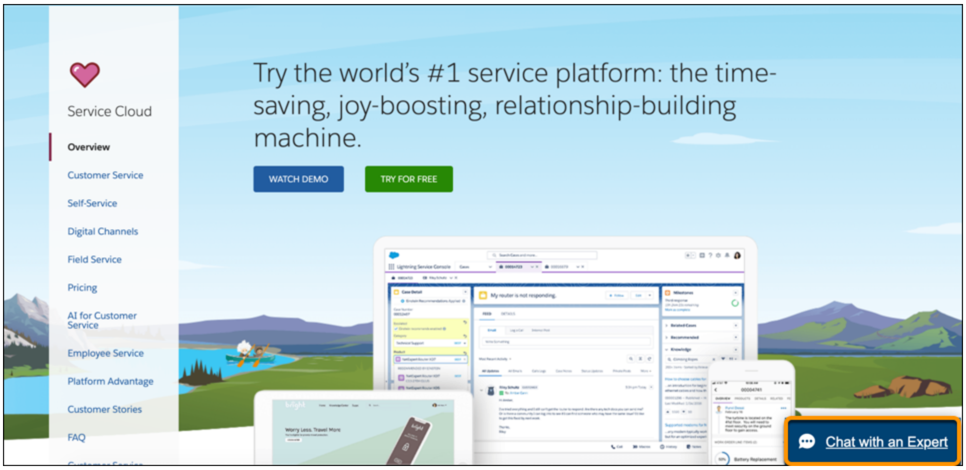
- Create an article

- Published articles—create a draft version or archive the article
- Draft articles—publish, edit, or delete
- Archived articles—restore or delete
- Bulk actions—archive, assign, publish, restore, submit for translation, and delete drafts or archived articles

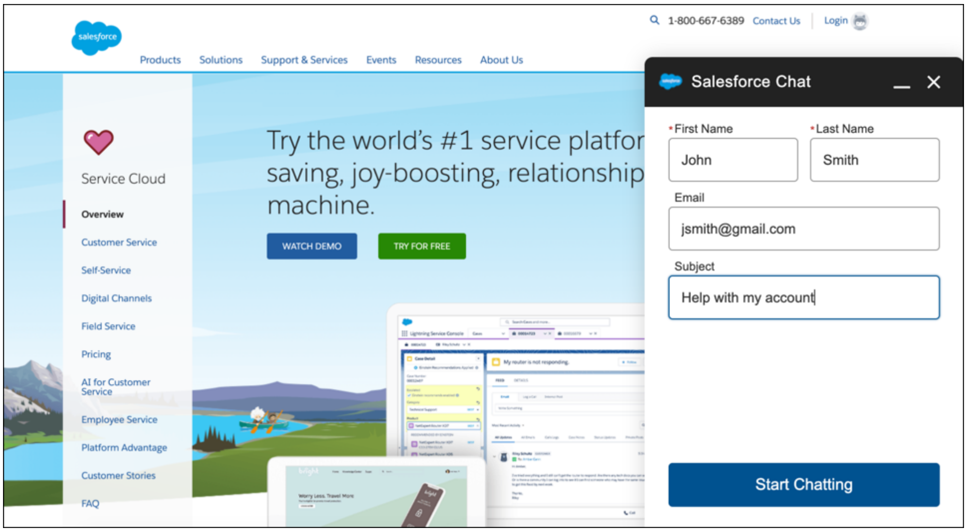
Embedded Chat for Fast, Snappy Service

Embedded chat allows customers to get quick answers to their questions by chatting with an agent while browsing your site. So, it's a great idea to add the Embedded chat widget to your website! Embedded chat uses the original Chat configuration, but in Lightning Experience you have a simpler setup on the Service Console.

What the customer sees: the chat widget button sits on your web page, and when someone has a question, they simply click the button to launch the chat.



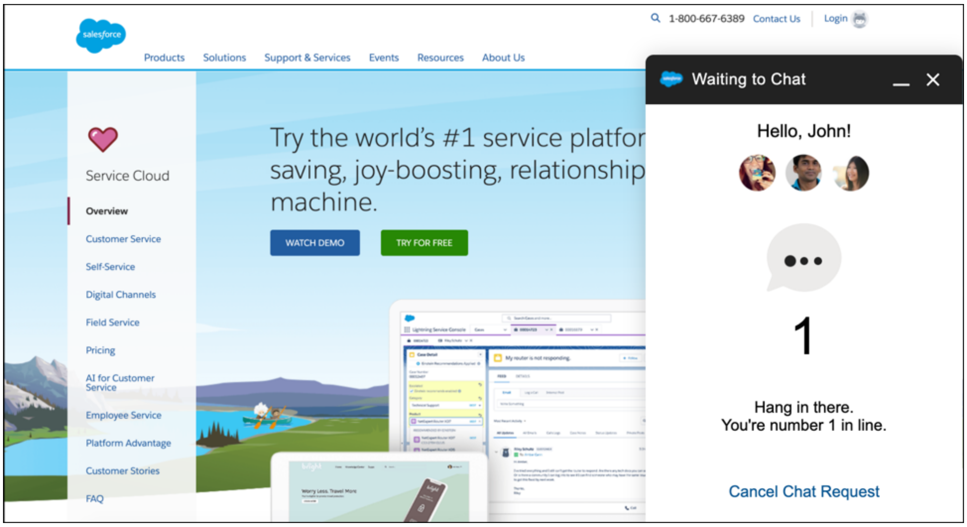
Customers fill out the brief pre-chat form, which helps agents gather basic information about the customer, like their contact information and their needs.



Salesforce Chat

*First Name	*Last Name
John	Smith
Email	
jsmith@gmail.com	
Subject	
Help with my account	

Customers can chat while viewing your web page, and they can minimize the chat window whenever it's in their way. The chat widget persists across your web pages so customers can keep browsing while chatting with an agent.

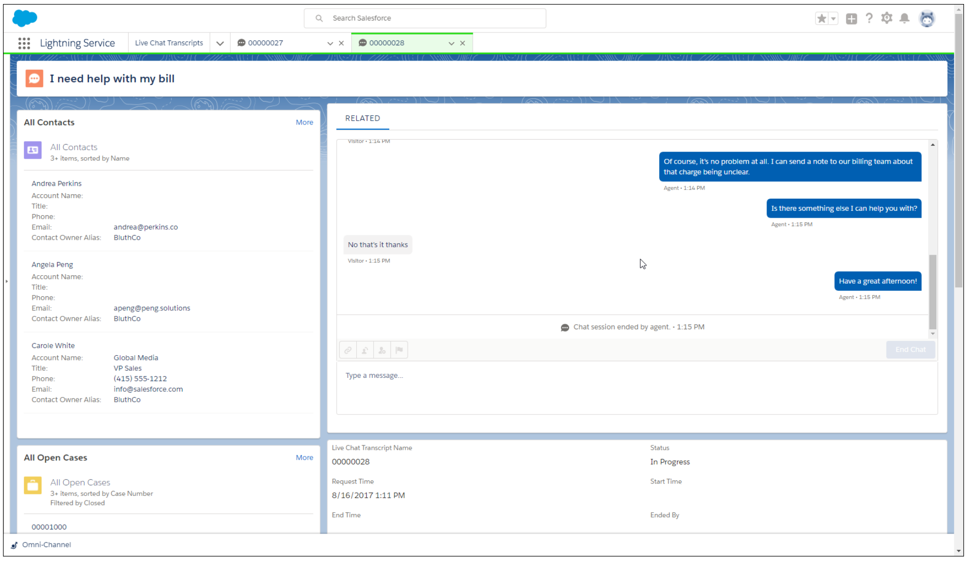


Hello, John!

1

Hang in there.
You're number 1 in line.

This example shows what support agents see when they chat with customers via Embedded chat. Embedded chat sits directly in the Service Console, allowing a quick and easy response to customers.



I need help with my bill

All Contacts

All Contacts
3+ items, sorted by Name

Andrea Perkins
Account Name:
Title:
Phone:
Email: andrea@perkins.co
Contact Owner Alias: BluthCo

Angela Peng
Account Name:
Title:
Phone:
Email: apeng@peng.solutions
Contact Owner Alias: BluthCo

Carole White
Account Name: Global Media
Title: VP Sales
Phone: (415) 555-1212
Email: info@salesforce.com
Contact Owner Alias: BluthCo

All Open Cases

All Open Cases
3+ items, sorted by Case Number
Filtered by Closed

00001000

Visitor • 1:14 PM

Of course, it's no problem at all. I can send a note to our billing team about that charge being unclear.
Agent • 1:14 PM

Is there something else I can help you with?
Agent • 1:15 PM

No that's it thanks
Visitor • 1:15 PM

Have a great afternoon!
Agent • 1:15 PM

Chat session ended by agent. • 1:15 PM

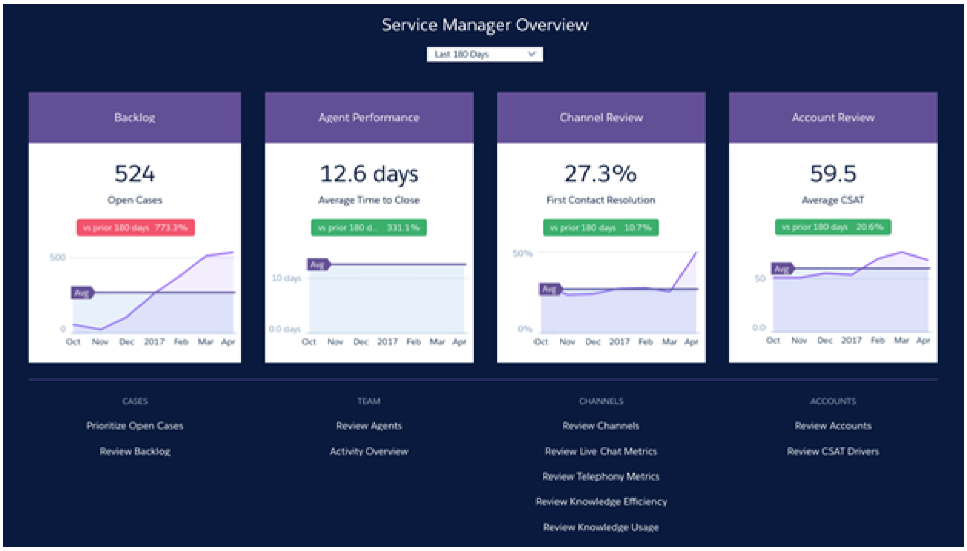
Type a message...

Live Chat Transcript Name: 00000028
Status: In Progress
Request Time: 8/16/2017 1:11 PM
Start Time:
End Time:
Ended By:

Analyze the Service Analytics App

Help your support team use data to uncover key insights and drive the success of your service business. The Service Analytics app shares best-practice key performance indicators (KPIs) about your service data in a single place. Even better, the app tailors this information by role, so everyone on the team gets the right information at the right time.

Using the prebuilt dashboards in the Service Analytics app, service managers can quickly view average case closing times, customer satisfaction, and trending, historical, and peer benchmarks. They can also get insight into the team's use of knowledge articles to resolve cases, and other data so they can quickly take appropriate actions.



Another set of dashboards—called sidebars—are specifically for support agents. For a given case, an agent can view customer history, number of cases, and CSAT. Ensure that agents have fast access to this information by embedding these sidebars on key Salesforce pages, such as the service console.

Sounds complicated and time-consuming to set up? Most of the hard work is already done for you! We built the complex queries, formulas, and ratios that draw from your service and sales data, and assembled them into easy-to-read visualizations. All you do is run a built-in configurator and answer a few questions about the data and fields you'd like to see. Woohoo!

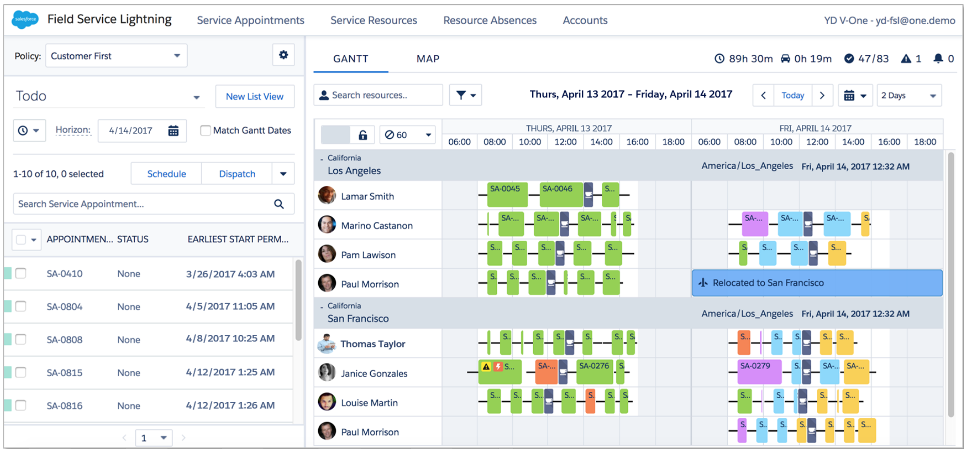
Get Out There with Field Service Lightning

Field Service Lightning is a mobile-friendly field service hub for your service team. Your agents can easily connect with people who work at customer sites, or who work on the go in delivery vehicles. Field Service Lightning helps everyone on the team make smarter decisions with visibility into operational performance and key performance indicators.

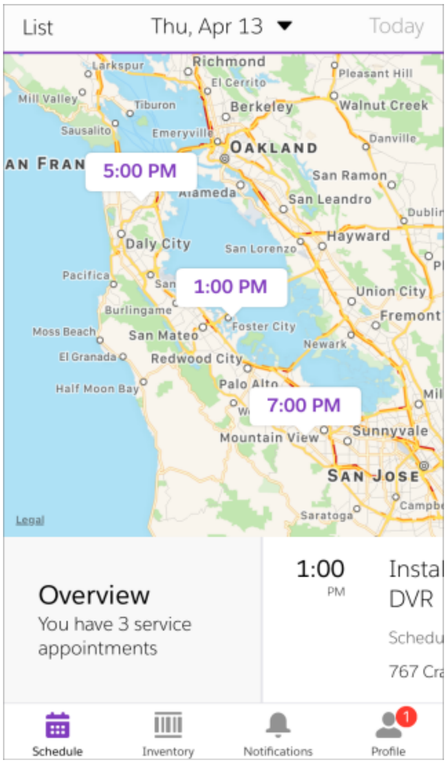
Running a field service business means managing many moving parts. With Field Service Lightning, your agents, dispatchers, and field technicians get the tools to manage work orders, service resources, and scheduling. But what about specifics? How about:

- Create service resource records that represent your field service technicians and add details about their skills, service territories, and availability.
- Set up multilevel service territories that represent the regions where your technicians work.

- Track your product inventory and service vehicle locations.
- Schedule one-time or recurring appointments for customers and add details about technician preference and required skills.
- Standardize your business's field service tasks with maintenance plans and templates.
- Keep customers informed about service progress with on-site service reports.



APPOINTMEN...	STATUS	EARLIEST START PERM...
SA-0410	None	3/26/2017 4:03 AM
SA-0804	None	4/5/2017 11:05 AM
SA-0808	None	4/8/2017 10:25 AM
SA-0815	None	4/12/2017 1:25 AM
SA-0816	None	4/12/2017 1:26 AM



Overview
You have 3 service appointments

1:00 PM
Instal
DVR
Schedu
767 Cr

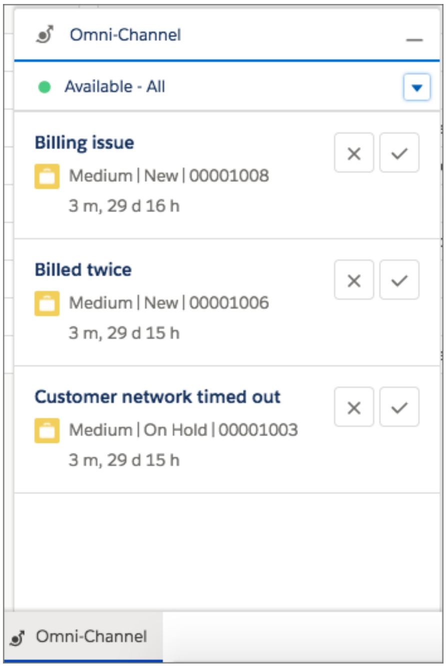
Omni-Channel Knows All

Omni-Channel helps your service center route any type of incoming work item—including cases, chats, or leads—to your most qualified, available agents. It's flexible and customizable, and you can configure it without writing code. Pretty rad!

Use Omni-Channel to manage the priority of work items for agents and balance the distribution of assignments. You can also define which agents work on different types of assignments, such as leads or sales inquiries, and other assignments that help with support questions.

Omni-Channel routes all these assignments to the correct agents automatically, which your agents will love. They no longer have to pick and choose work assignments from a queue, which saves everyone time, effort, and brainpower.

In Lightning Experience, the Omni-Channel utility delivers work right to your Lightning Service Console users via requests routed through Omni-Channel. Console users can set their presence status and accept or decline work depending on your Omni-Channel settings.



Billing issue
Medium | New | 00001008
3 m, 29 d 16 h

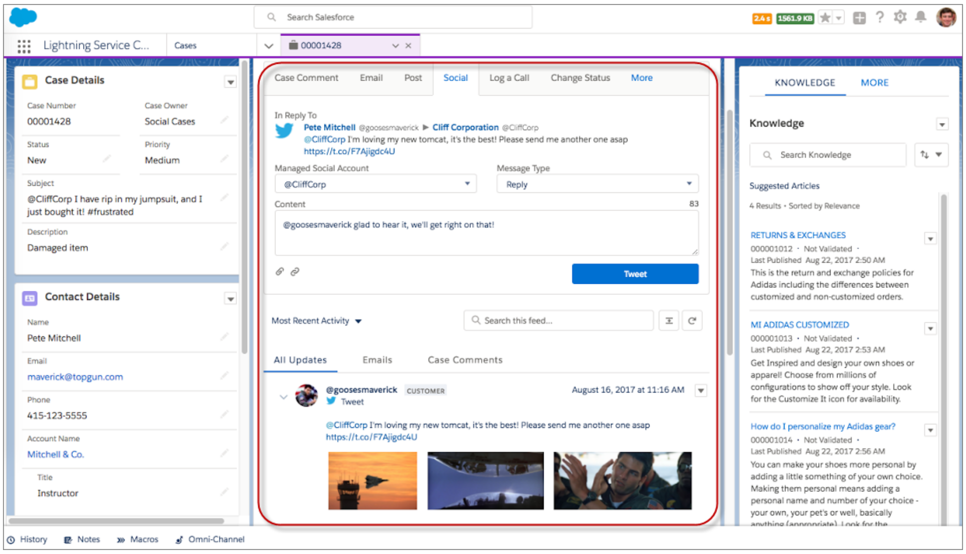
Billed twice
Medium | New | 00001006
3 m, 29 d 15 h

Customer network timed out
Medium | On Hold | 00001003
3 m, 29 d 15 h

Social Customer Service

Help your support team deliver personalized customer service from one of your most important channels—social media. Social Customer Service integrates with Twitter, Facebook, Instagram, and YouTube to deliver customer conversations directly to your agents. Setup is easier than ever in Lightning Experience, including Social Business Rules to quickly define how social cases are handled.

Support agents get the tools to manage cases originating from social media and engage customers by responding directly in social networks.



Search Salesforce

Lightning Service C... | Cases | 00001428

Case Details

Case Number	Case Owner
00001428	Social Cases
Status	Priority
New	Medium

Subject
@CliffCorp I have rip in my jumpsuit, and I just bought it! #frustrated

Description
Damaged item

Contact Details

Name
Pete Mitchell

Email
maverick@topgun.com

Phone
415-123-5555

Account Name
Mitchell & Co.

Title
Instructor

Case Comment | Email | Post | Social | Log a Call | Change Status | More

In Reply To
Pete Mitchell @goosesmaverick ▶ **Cliff Corporation** @CliffCorp
@CliffCorp I'm loving my new tomcat, it's the best! Please send me another one asap
https://t.co/F7Ajigdc4U

Managed Social Account: @CliffCorp
Message Type: Reply

Content 83
@goosesmaverick glad to hear it, we'll get right on that!

Tweet

Most Recent Activity | Search this feed...

All Updates | Emails | Case Comments

@goosesmaverick CUSTOMER — Tweet — August 16, 2017 at 11:16 AM
@CliffCorp I'm loving my new tomcat, it's the best! Please send me another one asap
https://t.co/F7Ajigdc4U

KNOWLEDGE | MORE

Knowledge

Search Knowledge

Suggested Articles
4 Results • Sorted by Relevance

RETURNS & EXCHANGES
000001012 • Not Validated •
Last Published Aug 22, 2017 2:50 AM
This is the return and exchange policies for Adidas including the differences between customized and non-customized orders.

MI ADIDAS CUSTOMIZED
000001013 • Not Validated •
Last Published Aug 22, 2017 2:53 AM
Get Inspired and design your own shoes or apparel! Choose from millions of configurations to show off your style. Look for the Customize It icon for availability.

How do I personalize my Adidas gear?
000001014 • Not Validated •
Last Published Aug 22, 2017 2:56 AM
You can make your shoes more personal by adding a little something of your own choice. Making them personal means adding a personal name and number of your choice - your own, your pet's or well, basically anything (appropriate). Look for the

History | Notes | Macros | Omni-Channel

Inbound and outbound social posts appear as items in the case feed, making it easy to follow conversations. Attachments to posts are also viewable right in the feed. Agents can respond to posts using the Social action in the publisher and include an attachment or a Direct Message prompt. They can also like conversations, reply privately to customers, hide comments on Facebook, and delete conversations managed by your company's social accounts.

CHAPTER 6 Explore More of Lightning Experience

In this chapter ...

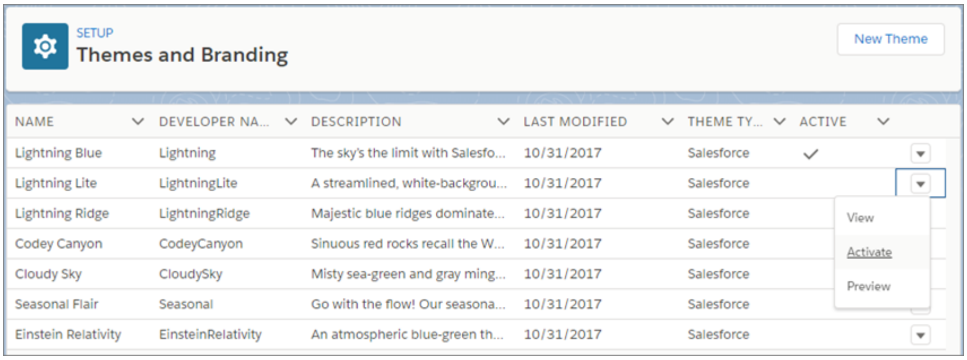
Opportunities and leads aren't the only places we've made improvements. There are slick new features to discover elsewhere, like custom brand images and colors in your org, the new Home page, revamped object home pages and enhanced list views, and enhanced Chatter and activities options.

Custom Themes and Branding

Rally your team around your brand and important initiatives, like a new product launch, with a set of custom brand images and colors in your org. We call them *themes*. You can choose one of Salesforce's built-in themes or create your own custom themes with just a few clicks.

Theme and brand-based colors show up in various places in the user interface, including page backgrounds, the global header, navigation bar, tabs, and buttons. You can also use a theme's brand image on the Lightning Experience loading page. By default, orgs use the built-in Lightning Blue theme.

From Setup, enter `Themes and Branding` in the Quick Find box, then select **Themes and Branding**. You can view, preview, and activate an existing theme or click **New Theme** to create your own.



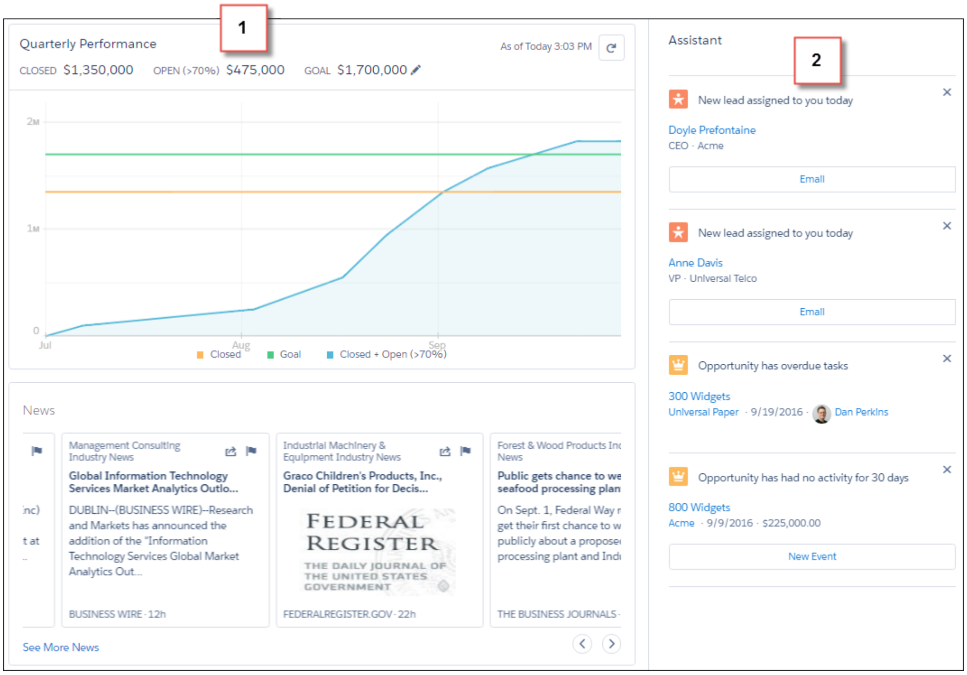
NAME	DEVELOPER NA...	DESCRIPTION	LAST MODIFIED	THEME TY...	ACTIVE
Lightning Blue	Lightning	The sky's the limit with Salesfo...	10/31/2017	Salesforce	✓
Lightning Lite	LightningLite	A streamlined, white-backgrou...	10/31/2017	Salesforce	
Lightning Ridge	LightningRidge	Majestic blue ridges dominate...	10/31/2017	Salesforce	
Codey Canyon	CodeyCanyon	Sinuous red rocks recall the W...	10/31/2017	Salesforce	
Cloudy Sky	CloudySky	Misty sea-green and gray ming...	10/31/2017	Salesforce	
Seasonal Flair	Seasonal	Go with the flow! Our seasona...	10/31/2017	Salesforce	
Einstein Relativity	EinsteinRelativity	An atmospheric blue-green th...	10/31/2017	Salesforce	

Upload a brand image, and choose brand colors. Custom themes modify the brand color to make it accessible. However, you can override the accessible brand color. Optionally, upload a page background image, default banners, and avatars for groups and user profiles.

The Home Page

The Home page displays key items for each user's day. From the Home page, your users can manage their day, including viewing their quarterly performance summary and the most relevant tasks and updates. You can also use the Lightning App Builder to create custom Home pages that appear for different profiles.

Give your users access to opportunity details so that they can get the most out of the Home page.



CLOSED $1,350,000 OPEN (>70%) $475,000 GOAL $1,700,000

Management Consulting Industry News
Global Information Technology Services Market Analytics Outlo...
DUBLIN--(BUSINESS WIRE)--Research and Markets has announced the addition of the "Information Technology Services Global Market Analytics Out...
BUSINESS WIRE · 12h

Industrial Machinery & Equipment Industry News
Graco Children's Products, Inc., Denial of Petition for Decis...
FEDERAL REGISTER THE DAILY JOURNAL OF THE UNITED STATES GOVERNMENT
FEDERALREGISTER.GOV · 22h

Forest & Wood Products Inc News
Public gets chance to we seafood processing plan
On Sept. 1, Federal Way r get their first chance to w publicly about a propose processing plant and Ind
THE BUSINESS JOURNALS ·

New lead assigned to you today
Doyle Prefontaine
CEO · Acme

New lead assigned to you today
Anne Davis
VP · Universal Telco

Opportunity has overdue tasks
300 Widgets
Universal Paper · 9/19/2016 · Dan Perkins

Opportunity has had no activity for 30 days
800 Widgets
Acme · 9/9/2016 · $225,000.00

Performance Chart (1)

The performance chart displays data based on opportunities belonging to the user or the user's sales team. Only opportunities for the current sales quarter that are closed or open with a probability over 70% are displayed. Multicurrency is supported in the performance chart.

Curious about the numbers at the top of the chart? Here's what they mean.

- Closed—The sum of a user's closed opportunities.
- Open (>70%)—The sum of a user's open opportunities with a probability over 70%. The blue line in the chart is the combined total of the closed opportunities and open opportunities with a probability over 70%.
- Goal—A user's customizable sales goal for the quarter. This field is specific to the performance chart and has no impact on forecast quotas or any other type of goals. Click ✎ to set the goal.

Assistant (2)

The Assistant shows your users things they need to address, including new leads and activities related to opportunities.

Items in the Assistant appear in the following order:

- Leads assigned to you today
- Opportunities with overdue tasks
- Opportunities with no activity in 30 days
- Opportunities with no open activity
- Overdue opportunities

  Opportunities are overdue if they're still open after the Close Date. These updates stop appearing if it's been over 8 days since the Close Date.

If your users don't have access to activities on opportunities or if the opportunity pipeline is off, they instead see opportunities that have close dates over the next 90 days.

Note: In Salesforce Classic, the Home page has a Chatter feed. In Lightning Experience, that feed isn't there by default. You can customize the page to add a feed, or access Chatter on the navigation bar. When you want to access Chatter on a record, go to the Chatter tab.

The Assistant doesn't show tasks due today or overdue tasks that aren't tied to an opportunity. The Today's Tasks component is an alternative that's available on the Home page, and it shows a list of your tasks due today.

To populate the performance chart, Top Deals, and the Assistant, users must have:

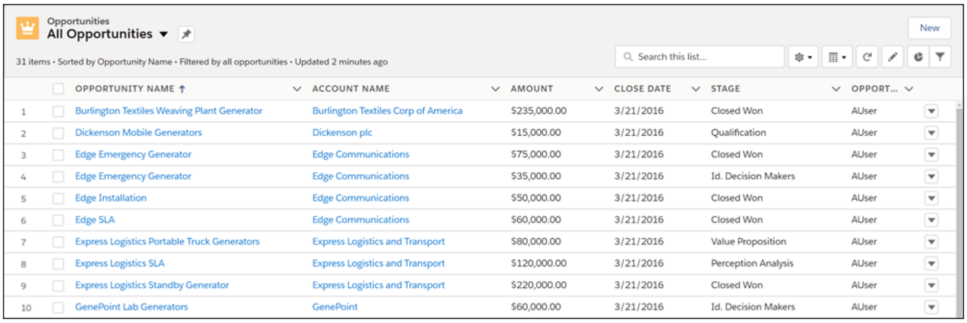
Table 1: Required Permissions for Home Features

Permission or Setting	Performance Chart	Key Deals	Assistant
Read access to the Opportunity object and sharing access to relevant opportunities	✓	✓	✓
Read access to the Opportunity object's Amount field	✓	✓	
Read access to the Opportunity object's Probability field	✓		
"Run Reports" user permission enabled for users	✓		

Permission or Setting	Performance Chart	Key Deals	Assistant
Closed opportunities or open opportunities with a probability over 70% during the current fiscal quarter	✓		
Read access to the Lead object			✓

Object Home Pages and List Views

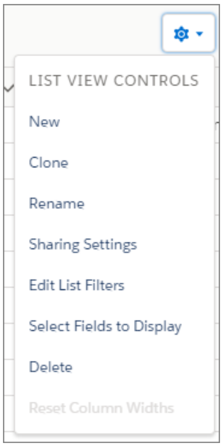
Salesforce Classic has a separate page for an object's home and another for its list views. In Lightning Experience, we've combined them!



31 items • Sorted by Opportunity Name • Filtered by all opportunities • Updated 2 minutes ago

	OPPORTUNITY NAME ↑	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORT...
1	Burlington Textiles Weaving Plant Generator	Burlington Textiles Corp of America	$235,000.00	3/21/2016	Closed Won	AUser
2	Dickenson Mobile Generators	Dickenson plc	$15,000.00	3/21/2016	Qualification	AUser
3	Edge Emergency Generator	Edge Communications	$75,000.00	3/21/2016	Closed Won	AUser
4	Edge Emergency Generator	Edge Communications	$35,000.00	3/21/2016	Id. Decision Makers	AUser
5	Edge Installation	Edge Communications	$50,000.00	3/21/2016	Closed Won	AUser
6	Edge SLA	Edge Communications	$60,000.00	3/21/2016	Closed Won	AUser
7	Express Logistics Portable Truck Generators	Express Logistics and Transport	$80,000.00	3/21/2016	Value Proposition	AUser
8	Express Logistics SLA	Express Logistics and Transport	$120,000.00	3/21/2016	Perception Analysis	AUser
9	Express Logistics Standby Generator	Express Logistics and Transport	$220,000.00	3/21/2016	Closed Won	AUser
10	GenePoint Lab Generators	GenePoint	$60,000.00	3/21/2016	Id. Decision Makers	AUser

What's that gear menu for? It contains options for managing your list views.



In the charts panel ( ), you can change the type of chart or create a new one. Click to add, set, or remove filters.

The name of the recent records list that displays by default on the home page for standard and custom objects is different in Lightning Experience. It's called Recently Viewed. There's another list view available for most standard objects that has the object type specified in the name. For example, the list view for accounts is called Recently Viewed Accounts. These two list views, Recently Viewed and Recently Viewed `Object`, show the same records. Neither list is deletable, filterable, or editable.

As an administrator, you can configure an object's Recently Viewed search layout for your users. The search layout controls what all users see when they land on that object's home page. In Lightning Experience, from Setup, find the object in the Object Manager, then scroll to its Search Layout related list and edit the Search Results search layout.

You can also go directly to an object's details page by selecting **Edit Object** from the Setup menu.

Considerations When Working with Object Home Pages and List Views

Some features and links that are available on object home pages in Salesforce Classic aren't available on comparable pages in Lightning Experience. You can still view, create, edit, and delete list views on these pages, but the steps you take for these tasks are different, and there are differences in how list views display and behave. Here are a few examples. The rest can be found in Lightning Experience Considerations.

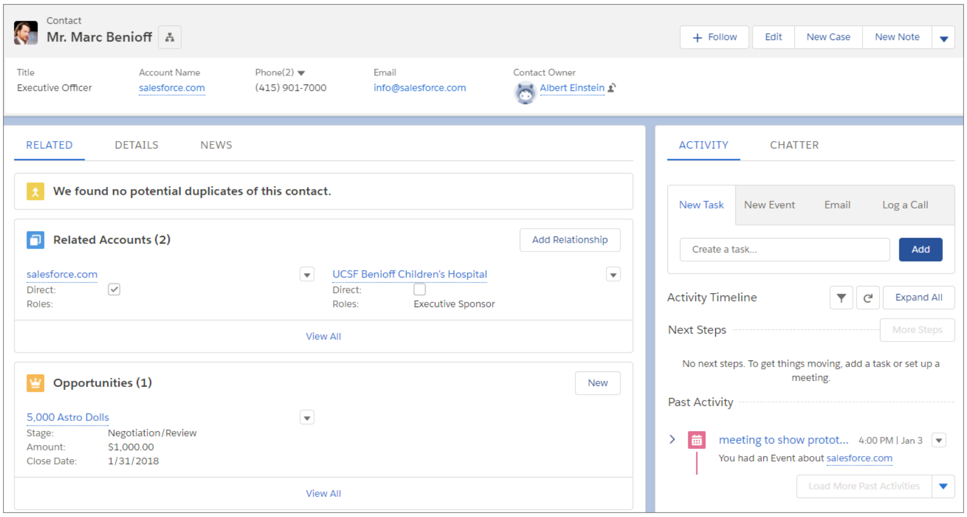
- Navigation through list views has changed.

- The list views dropdown menu shows up to 2,000 views. The menu loads views in batches of 50 as you scroll down. Recently viewed lists appear first, followed by other list views in alphabetical order. Recently viewed lists and the Search box appear only if there are 11 or more list views.
- Instead of an alphabet rolodex for list views, the List View Controls dropdown menu has auto-complete search that's enabled whenever there are more than 10 list views.
- You can change the columns that appear and their order by choosing **Select Fields to Display** from the List View Controls dropdown menu.
- Your users can reorder the columns of the Recently Viewed `Object` list in Salesforce Classic, and the changes are reflected in Lightning Experience.

How Other Objects are Organized in Lightning Experience

Opportunities and leads have special workspaces. The other objects have a different structure. Some elements, like related lists and the Activity and Chatter tabs, appear in different places than we saw on opportunities and leads.

For example, here's a contact page.


Contact
Mr. Marc Benioff

Title: Executive Officer
Account Name: salesforce.com
Phone(2): (415) 901-7000
Email: info@salesforce.com
Contact Owner: Albert Einstein

We found no potential duplicates of this contact.

Related Accounts (2)
salesforce.com
Direct: ✓
Roles:
UCSF Benioff Children's Hospital
Direct:
Roles: Executive Sponsor

Opportunities (1)
5,000 Astro Dolls
Stage: Negotiation/Review
Amount: $1,000.00
Close Date: 1/31/2018

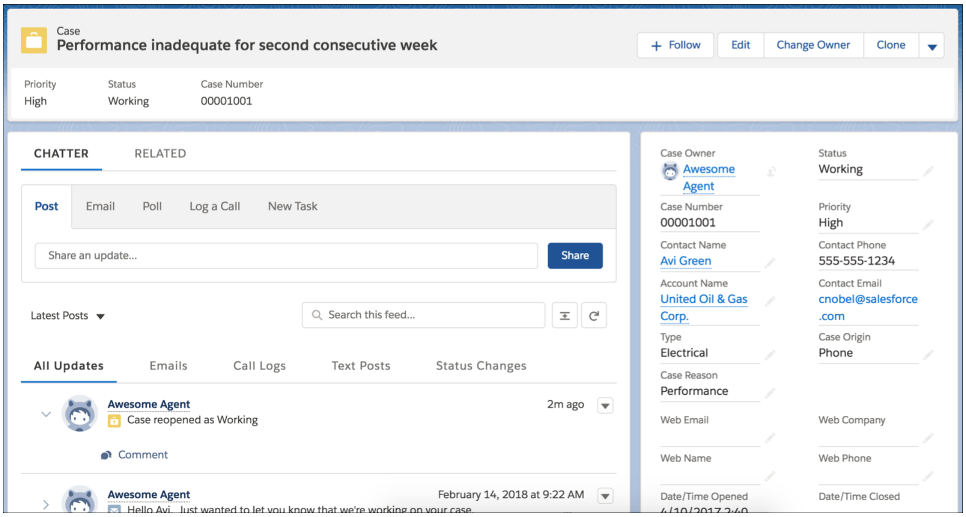
Activity Timeline
Next Steps
No next steps. To get things moving, add a task or set up a meeting.
Past Activity
meeting to show protot... 4:00 PM | Jan 3
You had an Event about salesforce.com

The highlights panel is in the same place across all objects. But here there's no Path, and the Activity and Chatter tabs are together in the right hand column. Related lists are in their own tab in the main part of the page, alongside the Details tab.

Some objects have special components that appear on their pages (for example, accounts, contacts, leads, and opportunities include News), but the overall structure remains the same across all objects.

Cases

Cases look slightly different than other records. Cases are feed-first and display a Chatter tab first, rather than record details or related information. The case feed helps support agents collaborate and work with cases faster. Details appear next to the feed. You don't see the Activity tab because this information displays directly in the Chatter case feed. In the Service Console app, cases display in a three-column page layout, with the Chatter publisher in the Feed tab.



Performance inadequate for second consecutive week

Priority: High
Status: Working
Case Number: 00001001

Awesome Agent — 2m ago
Case reopened as Working

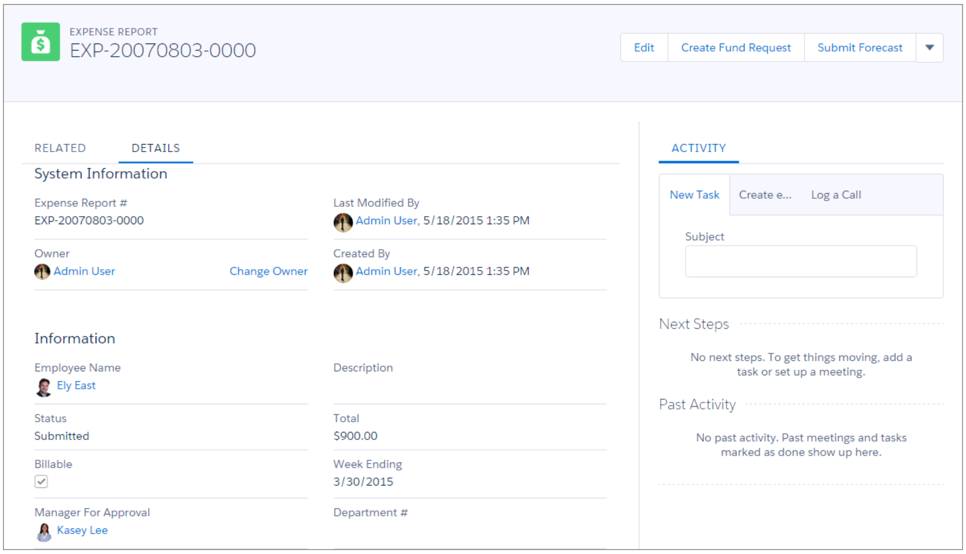
Awesome Agent — February 14, 2018 at 9:22 AM
Hello Avi, Just wanted to let you know that we're working on your case

Case Owner: Awesome Agent	Status: Working
Case Number: 00001001	Priority: High
Contact Name: Avi Green	Contact Phone: 555-555-1234
Account Name: United Oil & Gas Corp.	Contact Email: cnobel@salesforce.com
Type: Electrical	Case Origin: Phone
Case Reason: Performance	
Web Email	Web Company
Web Name	Web Phone
Date/Time Opened: 4/10/2017 2:40	Date/Time Closed

To fully benefit from the feed-first design on cases, recreate your Salesforce Classic case feed publisher actions in Lightning Experience. One benefit of recreating the publishers actions is that they appear on mobile devices, whereas the Salesforce Classic case feed publishers don't.

Custom Objects

Custom objects are supported in Lightning Experience, and custom object home pages contain the same standard elements as other objects, such as details, related lists, highlights panel, activities and a feed (Chatter).


EXPENSE REPORT
EXP-20070803-0000

Edit | Create Fund Request | Submit Forecast

RELATED | DETAILS

System Information

Expense Report # EXP-20070803-0000	Last Modified By Admin User, 5/18/2015 1:35 PM
Owner Admin User (Change Owner)	Created By Admin User, 5/18/2015 1:35 PM

Information

Employee Name Ely East	Description
Status Submitted	Total $900.00
Billable ✓	Week Ending 3/30/2015
Manager For Approval Kasey Lee	Department #

ACTIVITY

New Task | Create e... | Log a Call

Subject

Next Steps

No next steps. To get things moving, add a task or set up a meeting.

Past Activity

No past activity. Past meetings and tasks marked as done show up here.

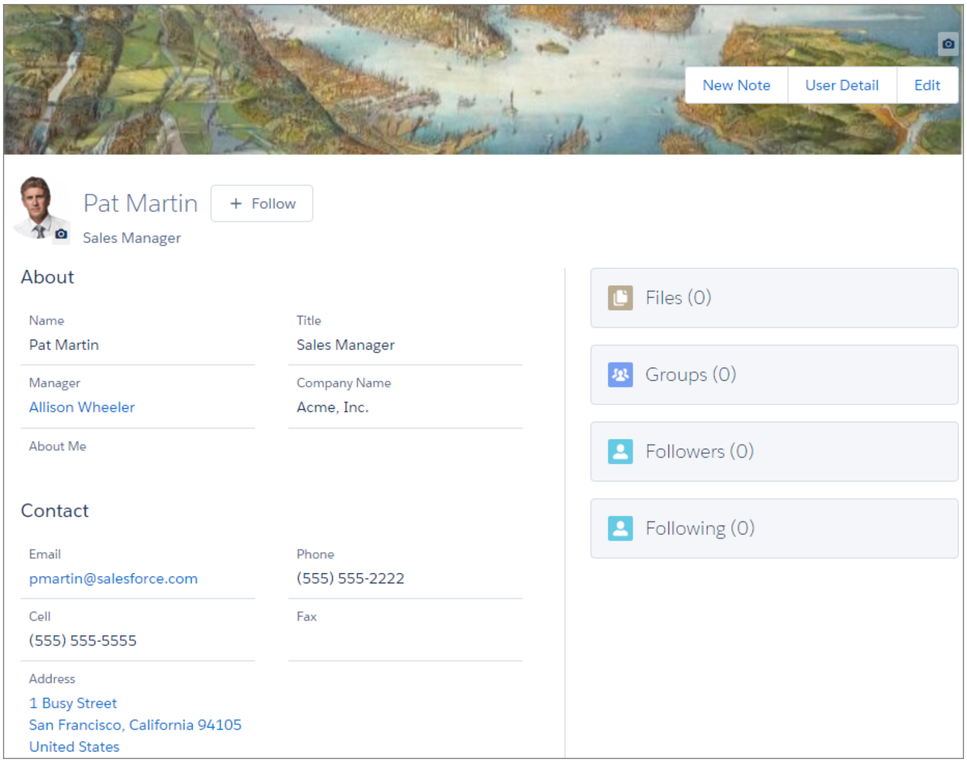
On this particular custom object, Expense Report, feed tracking hasn't been enabled. So, the Chatter tab, which you'd normally see next to Activity, doesn't appear.

Tip: On object home and record pages, you can select Edit Object from the Setup menu to go directly to that object's detail page.

User Profiles

The user profile page has been streamlined in Lightning Experience. From here, you can follow a user, see their details, edit their information (if you have permission), and see team members, files, groups, and other related information.

As with other objects, the pertinent details are in the main part of the page and the related lists are in the right-hand column. One unique thing about the user profile page is that the Google map that you see by default on address fields for other objects is disabled for users.



Pat Martin

Sales Manager

About

Name Pat Martin	Title Sales Manager
Manager Allison Wheeler	Company Name Acme, Inc.
About Me	

Contact

Email pmartin@salesforce.com	Phone (555) 555-2222
Cell (555) 555-5555	Fax
Address 1 Busy Street San Francisco, California 94105 United States	

Files (0)

Groups (0)

Followers (0)

Following (0)

Chatter in Lightning Experience

Wondering what's up with Chatter in Lightning Experience? You have options for displaying date and time stamps on Case feed items. More, better Chatter!

In Case feeds, you can now show users an absolute or relative date and time stamp on feed items. Mark Case posts and comments with an absolute date and time stamp like "January 7, 2020 at 12:15PM", or a relative period like "10 minutes ago".

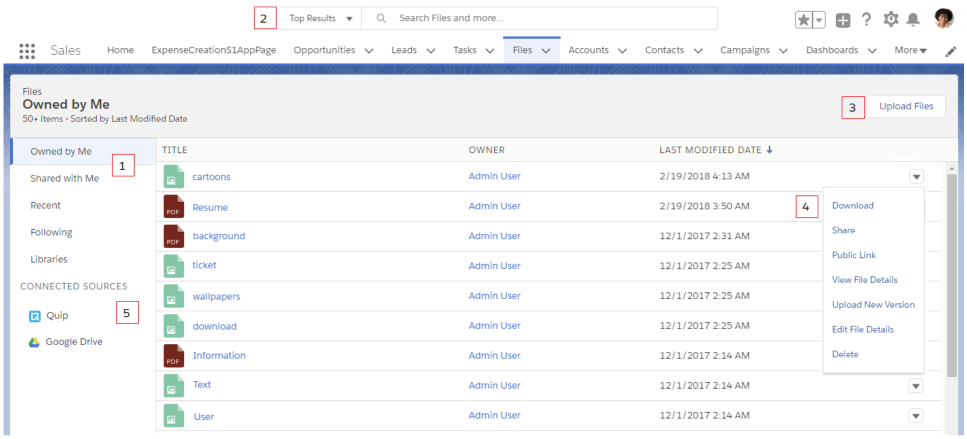
There's a new Chatter setting in Setup, **Show relative timestamp**. The relative time stamp is the default. To use an absolute time stamp, clear the checkbox. In a Case feed, the setting affects the date and time stamps on posts and comments. If you navigate to a feed item's detail view, the time stamp is always relative. When relative time is in effect in the feed, users can see the absolute date and time by hovering over the stamp.

File Management and Collaboration

Say goodbye to the days of juggling content across myriad types of Salesforce file management systems. In Lightning Experience, Salesforce Files unifies your users' files, documents, libraries, and attachments into a single place for easier file management. It also integrates access to files directly into the flow of your business, so users can get at the content they need from wherever they are in the app. Everyone in your org gets a welcome productivity boost with the integration of file sharing and content collaboration options into your sales and service processes.

Home Base for Managing Files

Files home is a powerful environment for working with and managing files. All file types are supported, including everything from Microsoft® PowerPoint presentations and Excel spreadsheets to Adobe® PDFs and image files. Audio and video files can be uploaded and downloaded, but can't be previewed or played. (An exception is Lightning Dialer voicemail messages, which are saved as files and can be played. Lightning Dialer is available for an extra cost as an add-on license.)



TITLE	OWNER	LAST MODIFIED DATE
cartoons	Admin User	2/19/2018 4:13 AM
Resume	Admin User	2/19/2018 3:50 AM
background	Admin User	12/1/2017 2:31 AM
ticket	Admin User	12/1/2017 2:25 AM
wallpapers	Admin User	12/1/2017 2:25 AM
download	Admin User	12/1/2017 2:25 AM
Information	Admin User	12/1/2017 2:14 AM
Text	Admin User	12/1/2017 2:14 AM
User	Admin User	12/1/2017 2:14 AM

- Refine the list of files using filters (1).
- Access libraries and the files contained in libraries. Libraries are a way to organize content and share content with users (1).
- Search (2) to find files.
- Upload files to Salesforce Files (3).
- Act on files directly from the file list (4).

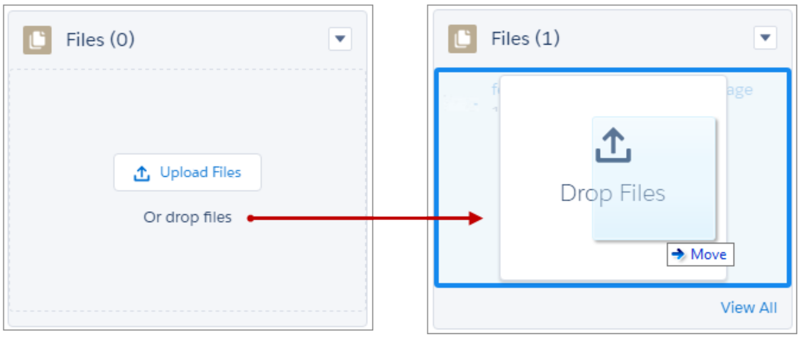
- If Files Connect is enabled for your org, the left navigation panel displays a section for external file sources, including Quip, Google Docs, and SharePoint (5).

Let's take a peek at a few of the other benefits your users get with Salesforce Files.

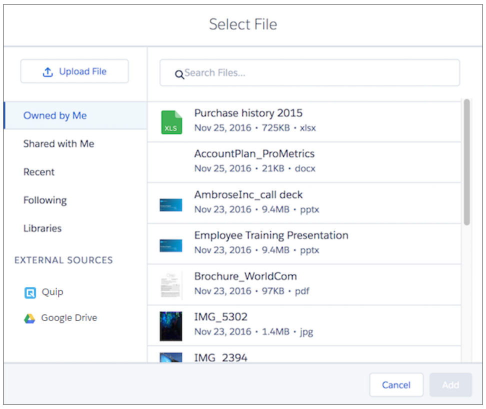
Manage Files on Records and Groups

Have your users created attachments for records or groups using the old Notes & Attachments related list? With the new Files related list, you can give everyone a better way to manage related content.

When someone adds an attachment to a record, it exists only in the context of that record. Users can't get to the attachment from anywhere else in Salesforce. Pretty limiting, right? That's why we developed the Files related list. When you include the Files related list on an object's page layout, a user can upload files from their computer to attach to a record. The uploaded file is automatically added to Salesforce Files so it's accessible from anywhere, not just the record at hand. Simply drag a file—or multiple files at once—onto the related list! Or click **Add Files** > **Upload Files** to navigate through the file system to the right place.



The Files related list also opens a door to all a user's files in Salesforce Files, including files in libraries and connected files in external repositories like Quip, Box, and Google Drive.



Select File

Upload File

Search Files...

Owned by Me
Shared with Me
Recent
Following
Libraries
EXTERNAL SOURCES
Quip
Google Drive

Purchase history 2015
Nov 25, 2016 • 725KB • xlsx

AccountPlan_ProMetrics
Nov 25, 2016 • 21KB • docx

AmbroseInc_call deck
Nov 23, 2016 • 9.4MB • pptx

Employee Training Presentation
Nov 23, 2016 • 9.4MB • pptx

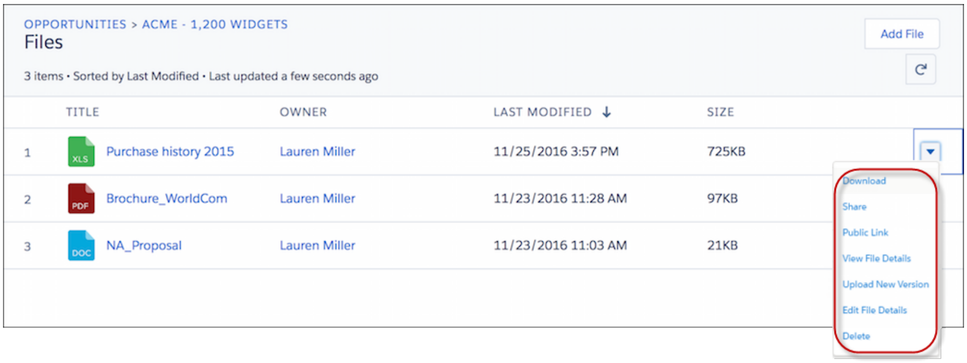
Brochure_WorldCom
Nov 23, 2016 • 97KB • pdf

IMG_5302
Nov 23, 2016 • 1.4MB • jpg

IMG_2394

Cancel Add

Drill in on the Files related list and all the rich features and flexibility of Salesforce Files are available for each file in the list. For example, upload a new version of a file, edit a file's details, or share a file with people, groups, and libraries.



OPPORTUNITIES > ACME - 1,200 WIDGETS
Files

Add File

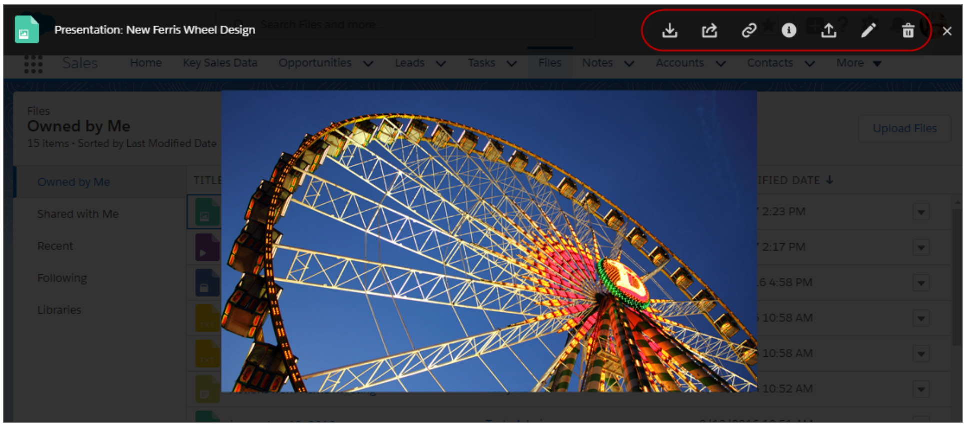
3 items • Sorted by Last Modified • Last updated a few seconds ago

	TITLE	OWNER	LAST MODIFIED ↓	SIZE
1	Purchase history 2015	Lauren Miller	11/25/2016 3:57 PM	725KB
2	Brochure_WorldCom	Lauren Miller	11/23/2016 11:28 AM	97KB
3	NA_Proposal	Lauren Miller	11/23/2016 11:03 AM	21KB

Preview Files

The beautiful file preview player in Lightning Experience provides a visually and functionally rich preview experience. Preview images are vector-based (rather than pixel-based), so the render quality is higher and doesn't degrade on high-resolution screens. And there's no need for your users to install Adobe Flash Player to preview files because the player isn't SWF-based.

The player includes controls for quickly downloading, sharing, or deleting the current file, uploading a new version of the file, editing file details, and generating a public link to the file. It also has full-screen mode and works with most presentation clickers, so reps can dazzle their audiences without needing to download anything.



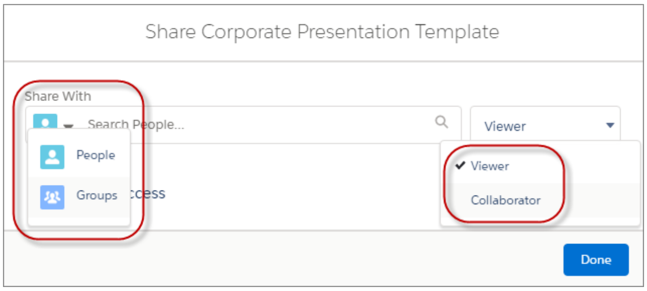
Share Files

To facilitate content collaboration, Salesforce Files embeds file sharing opportunities throughout your company's workflow. Users can access sharing options from Files home, Files related lists, the file preview player, file record detail pages, and Chatter feeds.

It's easy to share files with the right audience.

Specific people or groups

Share files with just the people or groups who should see them. Assign unique file permissions for each person or group. Viewers can view, download, and reshare the shared file (if the file allows resharing). Collaborators can do all that plus change file permissions, edit the file, and upload new versions.

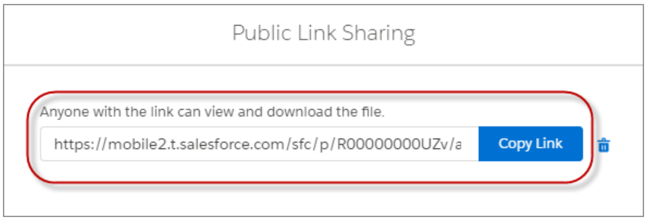


Broader audiences or records

Attach files to posts or comments in the main Chatter feed or to feeds on records.

Customers or people outside your company

Share a file with external audiences by creating a public link. A public link is an encrypted URL that you can send to people outside (or inside) your company, including leads, customers, partners, and coworkers. Recipients can view and download the file but they can't be collaborators. You can delete the link at any time, at which point it no longer provides access to the file.



Anyone with the link can view and download the file.

https://mobile2.t.salesforce.com/sfc/p/R00000000UZv/a

Considerations for Other Content Types

Some older Salesforce content types aren't fully supported or supported at all in Lightning Experience. Here are a few things to keep in mind. See Lightning Experience Considerations in Salesforce Help for the full details.

Attachments

Attachments on records aren't automatically available from Salesforce Files or the Files related list—because attachments aren't actually files. In Lightning Experience, users can continue to access attachments from the Notes & Attachments related list. But it's confusing to have content in two different lists on the same record. Fortunately you can convert attachments to files and eliminate the need for wasteful context switching. Install the Attachments to Files tool from AppExchange and kick off bulk jobs that convert attachments to Salesforce Files. Your users will thank you!

Documents

The Documents tab isn't making the trek to Lightning Experience. But there are several options for making files from Documents available in Salesforce Files, including uploading them directly, using third-party data export tools, or using an API-based tool. See Moving Documents to Salesforce Files in Salesforce Help for more information.

One exception is documents used in email templates as logo headers and attachments. These files need to stay in Documents for now.

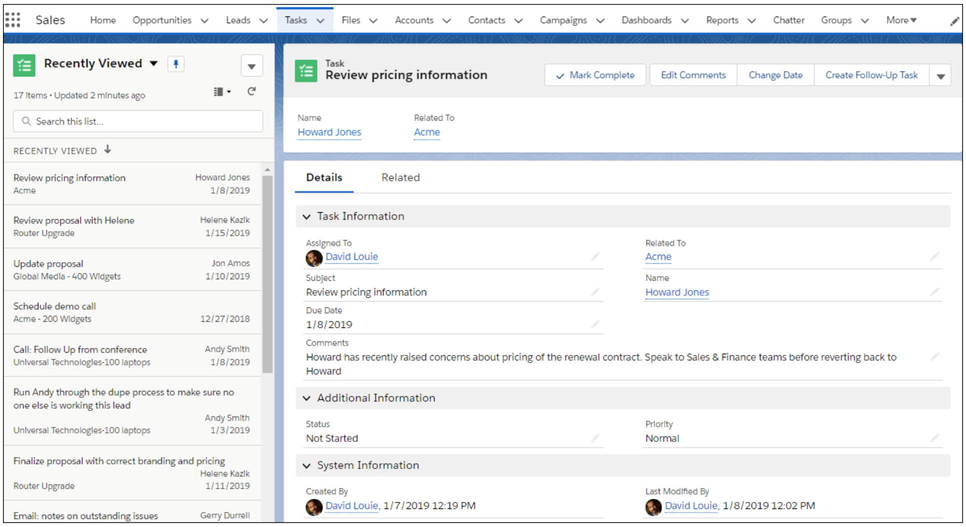
Salesforce CRM Content

Folders and files in libraries are available from Salesforce Files in Lightning Experience. Users can share and interact with them as with other files. Library admins can create and manage libraries in Lightning Experience too. Content deliveries aren't available in Lightning Experience, but link sharing is supported.

Tasks, Calendar, and Events

There's a better way to track the work you need to do to move deals ahead, with some nifty enhancements to tasks and calendars in Lightning Experience.

Tasks



17 Items • Updated 2 minutes ago

Review pricing information	Acme	Howard Jones 1/8/2019
Review proposal with Helene	Router Upgrade	Helene Kazik 1/15/2019
Update proposal	Global Media - 400 Widgets	Jon Amos 1/10/2019
Schedule demo call	Acme - 200 Widgets	12/27/2018
Call: Follow Up from conference	Universal Technologies-100 laptops	Andy Smith 1/8/2019
Run Andy through the dupe process to make sure no one else is working this lead	Universal Technologies-100 laptops	Andy Smith 1/3/2019
Finalize proposal with correct branding and pricing	Router Upgrade	Helene Kazik 1/11/2019
Email: notes on outstanding issues		Gerry Durrell

Task
Review pricing information

Name: Howard Jones
Related To: Acme

Assigned To: David Louie
Related To: Acme
Subject: Review pricing information
Name: Howard Jones
Due Date: 1/8/2019
Comments: Howard has recently raised concerns about pricing of the renewal contract. Speak to Sales & Finance teams before reverting back to Howard

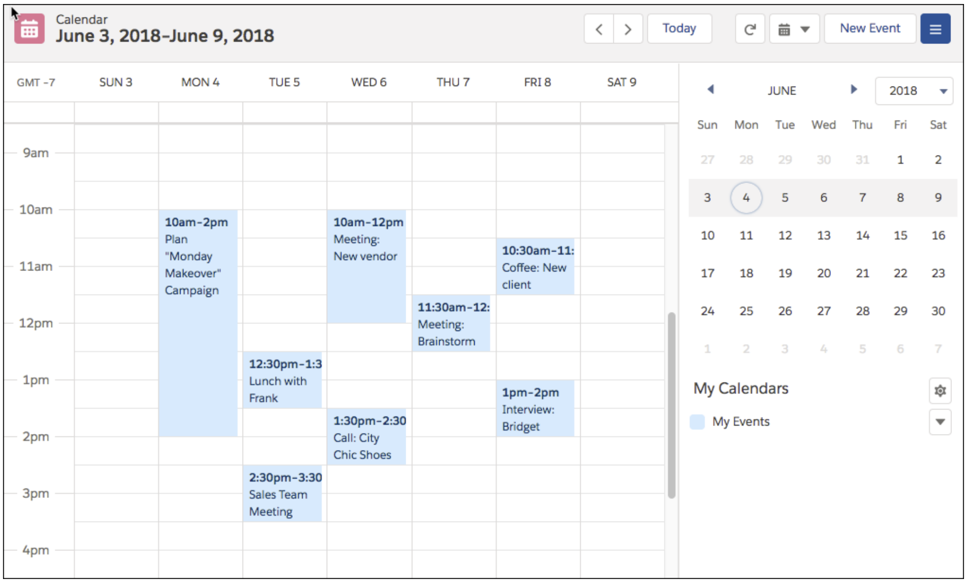
Status: Not Started
Priority: Normal

Created By: David Louie, 1/7/2019 12:19 PM
Last Modified By: David Louie, 1/8/2019 12:02 PM

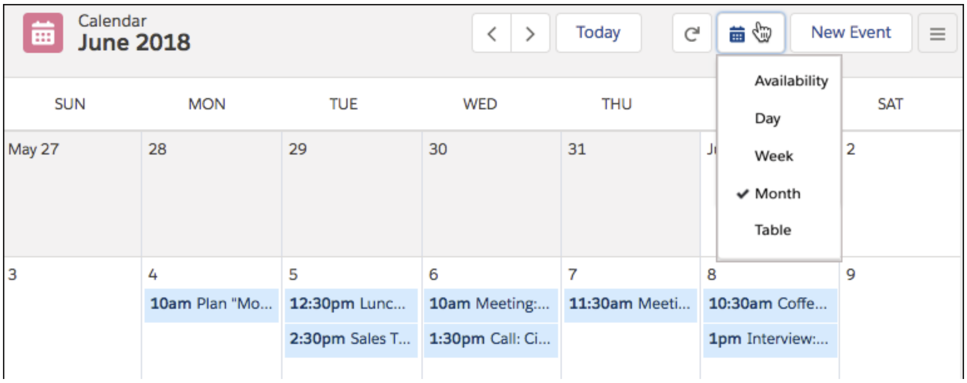
- View a list of all your open tasks, showing the opportunities, accounts, and other records they're related to. See and edit details right there on the same page.
- Switch to views of tasks you've delegated, open tasks, all overdue tasks, recently completed tasks, recently viewed tasks, recurring tasks, tasks due today, and unscheduled tasks. Quickly mark tasks complete directly from any tasks list view. Assign tasks to a queue so that you can share work more efficiently. Any member of a queue can complete a task when they have time.

Calendar

Here's the calendar week view:


Calendar
June 3, 2018–June 9, 2018
Today
New Event
GMT -7
SUN 3
MON 4
TUE 5
WED 6
THU 7
FRI 8
SAT 9
9am
10am
11am
12pm
1pm
2pm
3pm
4pm
10am–2pm Plan "Monday Makeover" Campaign
12:30pm–1:3 Lunch with Frank
2:30pm–3:30 Sales Team Meeting
10am–12pm Meeting: New vendor
1:30pm–2:30 Call: City Chic Shoes
11:30am–12: Meeting: Brainstorm
10:30am–11: Coffee: New client
1pm–2pm Interview: Bridget
JUNE
2018
Sun Mon Tue Wed Thu Fri Sat
My Calendars
My Events

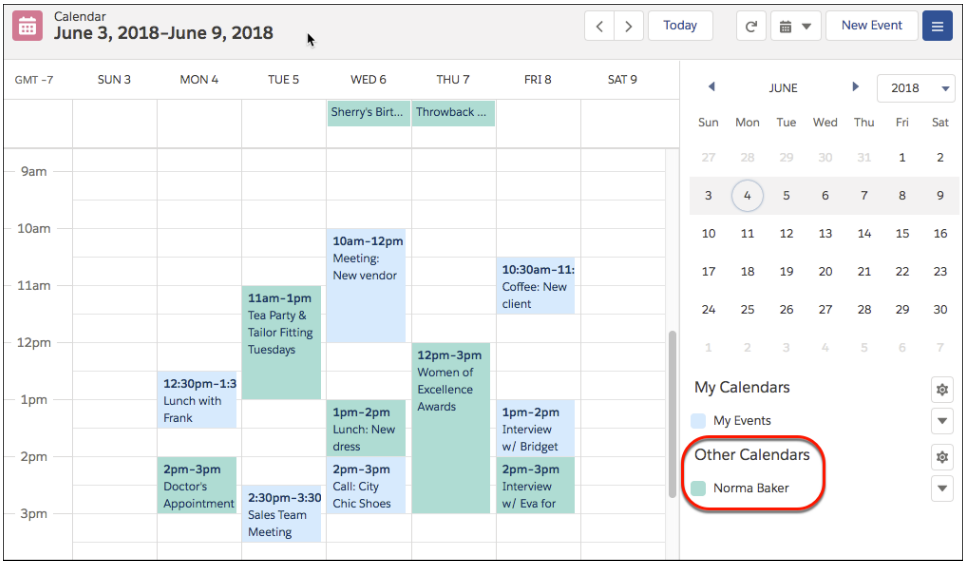
Or, change the calendar view to see your schedule for one day, or the whole month:



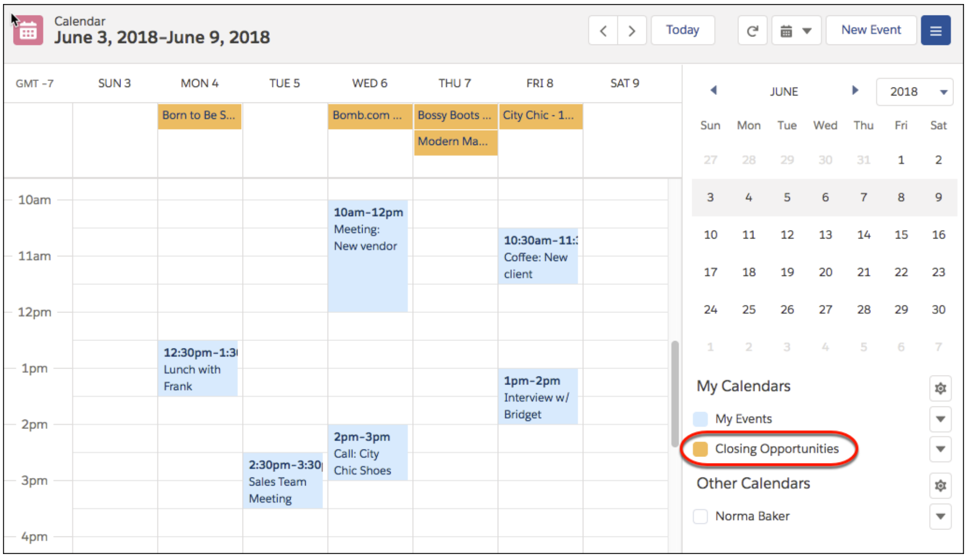
You can also see a list of events using the table view or find a time that works for multiple people in availability view.

When you view, create, and edit events in Lightning Experience, the calender is streamlined and easier to use. The calendar displays all events owned by a user, including events outside a user's business hours, in the time zone selected in your Salesforce settings.

To help you schedule meetings efficiently, you can take a peek at your coworkers' calendars, too. What you see depends on your admin-set sharing settings and the level of sharing access that coworkers give.

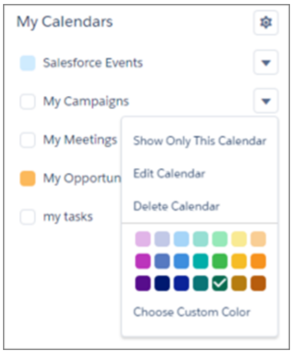

Calendar
June 3, 2018–June 9, 2018
Today
New Event
GMT -7
SUN 3
MON 4
TUE 5
WED 6
THU 7
FRI 8
SAT 9
Sherry's Birt...
Throwback ...
9am
10am
11am
12pm
1pm
2pm
3pm
10am–12pm Meeting: New vendor
10:30am–11: Coffee: New client
11am–1pm Tea Party & Tailor Fitting Tuesdays
12pm–3pm Women of Excellence Awards
12:30pm–1:3 Lunch with Frank
1pm–2pm Lunch: New dress
1pm–2pm Interview w/ Bridget
2pm–3pm Doctor's Appointment
2pm–3pm Call: City Chic Shoes
2pm–3pm Interview w/ Eva for
2:30pm–3:30 Sales Team Meeting
JUNE
2018
My Calendars
My Events
Other Calendars
Norma Baker

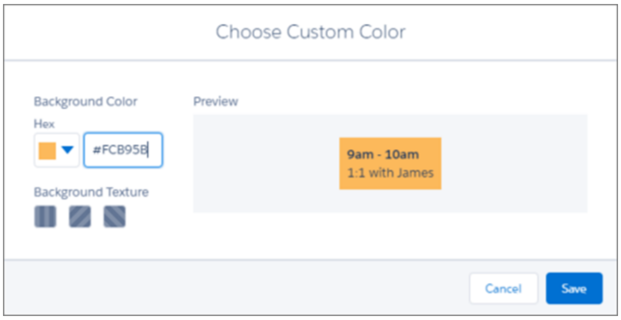
Let's say that you want to track campaigns, cases, retail store events, opportunities, and more. Just add calendars for all of these things. Choose a field to track on any standard or custom object. The calendar displays data in that field as calendar items.



Users can customize most calendars by applying a list view, and they can edit and delete calendars they've created.

You can distinguish calendars by color and texture.





Calendar views display up to 150 items, including items from calendars you create.

To work with attendees and generate invitations from your meetings, ask your admin about setting up the Lightning Sync feature.

CHAPTER 7 Reports and Dashboards in Lightning Experience

In this chapter ...

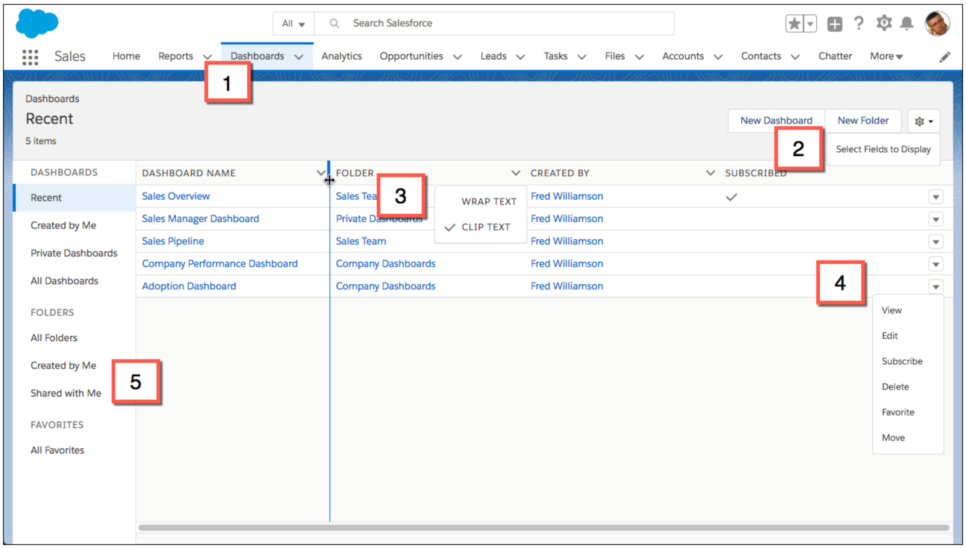
We've redesigned reports and dashboards for Lightning Experience, making them more interactive, easier to navigate, and easier to edit. Let's see what reports and dashboards in Lightning Experience can do!

Here's a quick overview:

- Get more information from interactive charts without needing to drill into reports (the charts look fabulous, too!)
- Create dashboards with more than three columns and column-spanning components using the Lightning dashboard editor
- Without launching the report builder, you can edit report filters, show or hide a chart, totals, and details
- Find, manage, and create reports and dashboards more easily via redesigned home pages
- Work with your reports and dashboards from Salesforce Classic in Lightning Experience

Report and Dashboard Home Pages

We've redesigned the Reports and Dashboards home pages in Lightning Experience so that your users can find and create reports and dashboards more easily.



DASHBOARD NAME	FOLDER	CREATED BY	SUBSCRIBED
Sales Overview	Sales Team	Fred Williamson	✓
Sales Manager Dashboard	Private Dashboards	Fred Williamson	
Sales Pipeline	Sales Team	Fred Williamson	
Company Performance Dashboard	Company Dashboards	Fred Williamson	
Adoption Dashboard	Company Dashboards	Fred Williamson	

Access your reports from the Reports tab, and your dashboards from the Dashboards tab (**1**).

You can build a new report or dashboard (**2**) right from this page by clicking **New Report** or **New Dashboard**. Create a folder by clicking **New Folder**. Customize the tab layout by clicking the gear icon and then clicking **Select Fields to Display**.

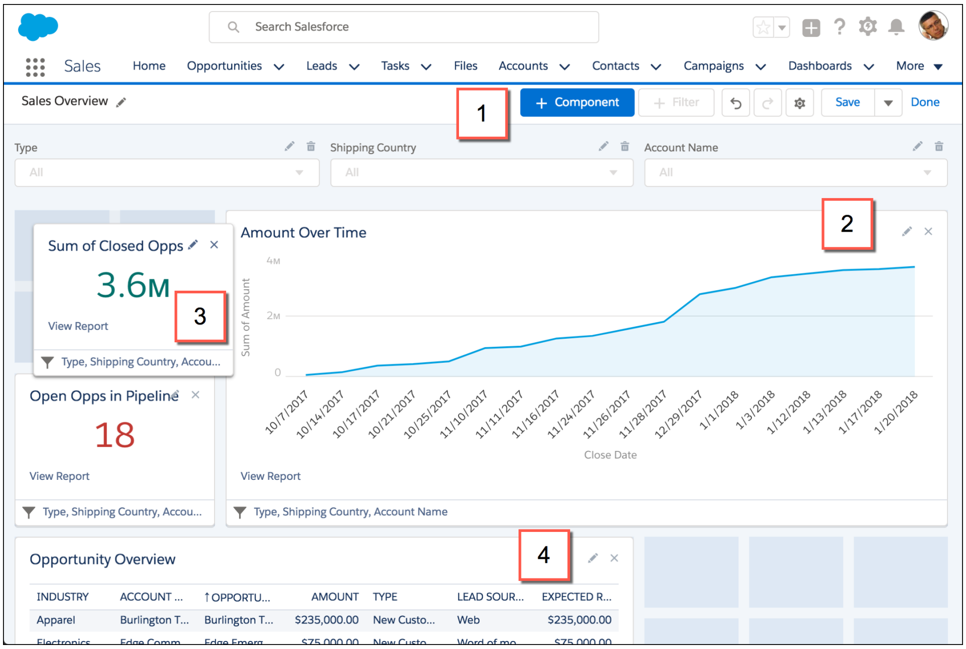
Click a column heading (**3**) to sort by name, folder, creator, or whoever last modified a report or dashboard. Click again to reverse the sort order. Resize columns by hovering over the border area and dragging the border. Choose whether text in a column clips or wraps from the column actions menu.

The more actions menu provides click access to whatever you want to do (**4**). It's the fastest way to edit, subscribe, or favorite. But, if all you want to do is view a report or dashboard, it's a little quicker to click the name instead!

Find a report or dashboard using filtered lists and folders (**5**). Filtered lists are the fastest way to find the report or dashboard you're looking for. For example, you can quickly find the Open Deals report you were reading last Friday in Recent. Folders let you group related reports or dashboards, so they're easy to find again later.

Create Beautiful Dashboards Using a Feature-Rich Editor

Lightning Experience introduces a new dashboard editor that'll usher in a whole new generation of dashboards.



INDUSTRY	ACCOUNT ...	↑OPPORTU...	AMOUNT	TYPE	LEAD SOUR...	EXPECTED R...
Apparel	Burlington T...	Burlington T...	$235,000.00	New Custo...	Web	$235,000.00
Electronics	Edge Comm...	Edge Emerg...	$75,000.00	New Custo...	Word of mo...	$75,000.00

Get started from the toolbar (**1**). From here, you can add components and filters, undo or redo edits, edit chart properties, and perform administrative tasks like saving. Unique to Lightning Experience, from the chart properties menu, you can customize chart colors and dashboard theme (background color).

To edit an already-placed dashboard component, click (**2**). You can choose a data-supplying source report, change chart types, edit the component title, and more. To remove a component, click **X**.

Arrange dashboard components more easily than ever by dragging them anywhere on the grid (**3**). Lightning Experience dashboards now feature more than three columns. They feature up to nine columns and unlimited rows. Dashboard components can span as many columns and rows as you like. Drag the corners and sides of a dashboard component to make it bigger or smaller. Charts, tables, and metrics all dynamically resize. Go ahead and get creative with your dashboard layout!

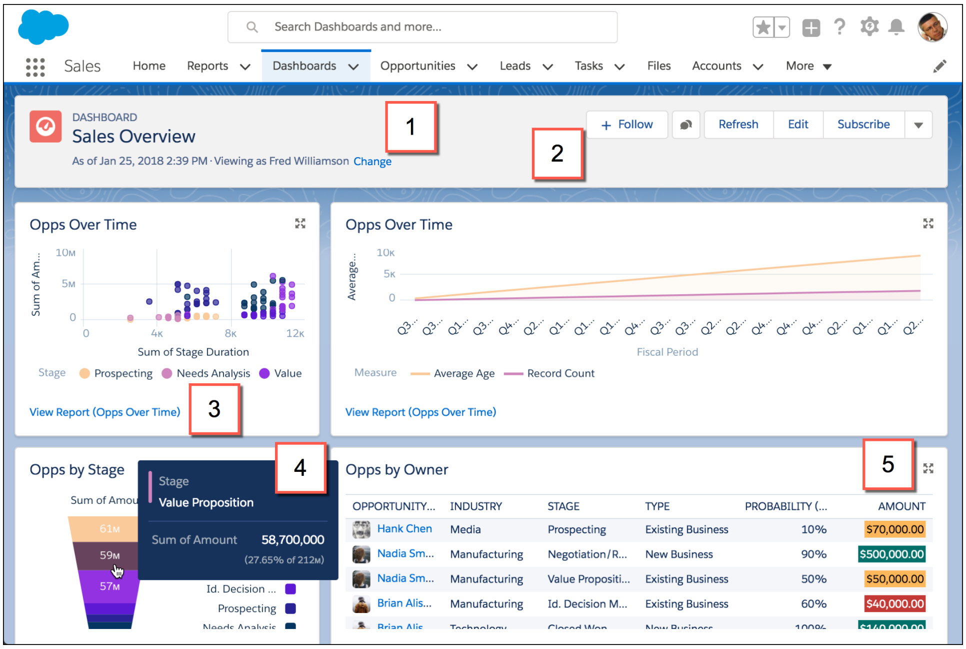
Put a better table in your dashboards [4]. The Lightning table is a new dashboard component that's only available in Lightning Experience. How much better is it? Here's what it offers.

- Up to 10 columns and 200 rows!
- Columns come from source reports' report type. That means that you don't have to add a field to the source report to show it in the table.
- Show Chatter photos and conditional highlighting.
- Lightning tables are dashboard filter compatible.
- Customize dashboard color palette and theme.

Lightning tables don't appear when you view the dashboard in Salesforce Classic.

Present and Share Information in Interactive Dashboards

Interactive dashboard components give viewers more information and link to data-supplying reports.



OPPORTUNITY...	INDUSTRY	STAGE	TYPE	PROBABILITY (...	AMOUNT
Hank Chen	Media	Prospecting	Existing Business	10%	$70,000.00
Nadia Sm...	Manufacturing	Negotiation/R...	New Business	90%	$500,000.00
Nadia Sm...	Manufacturing	Value Propositi...	Existing Business	50%	$50,000.00
Brian Alis...	Manufacturing	Id. Decision M...	Existing Business	60%	$40,000.00
Brian Alis...	Technology	Closed Won	New Business	100%	$140,000.00

View dashboards as other people (**1**). If properties are set to allow you to choose whom you view the dashboard as, click **Change** to see the dashboard as someone else.

Refresh, edit, and collaborate (**2**). Perform common tasks, like refreshing dashboard data, from the toolbar.

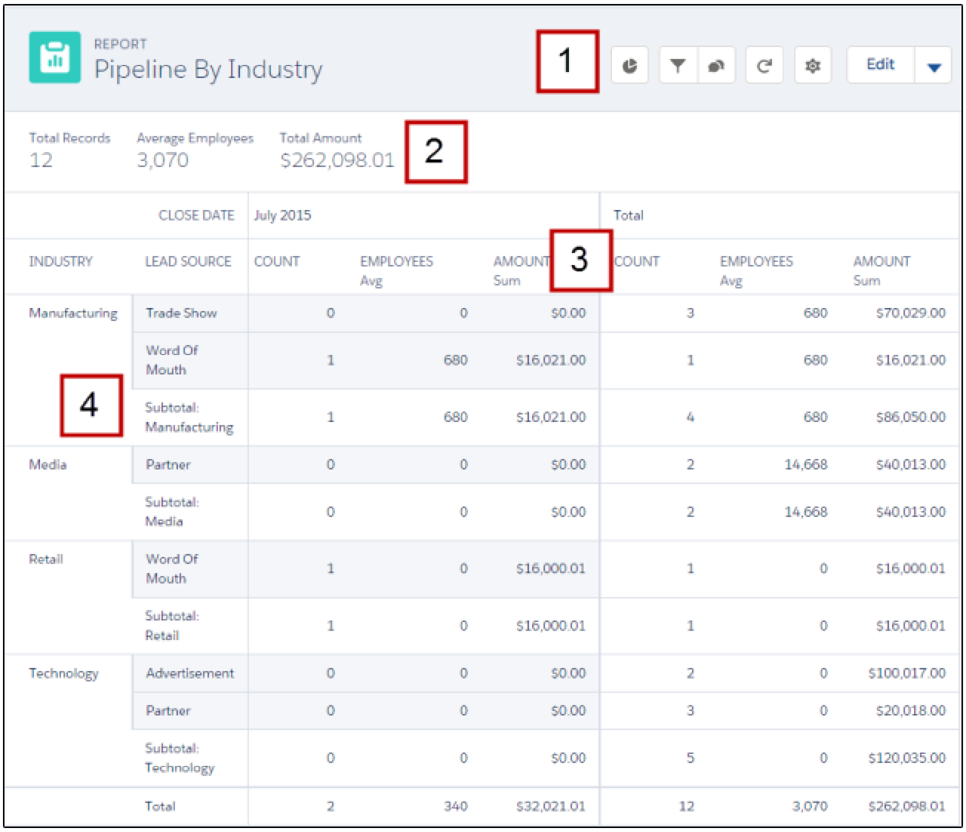
View source reports (**3**). To view a component's source report, click **View Report**. When it opens, dashboard filters get applied to the source report.

Hover over charts to see more details (**4**). Click a chart segment or datapoint to open a filtered source report.

Expand dashboard components to see a larger version of it (**5**). If a chart is a little too small to make out an important detail, expand it.

Get More Information from Interactive Charts and Filters on the Report Run Page

In Lightning Experience, new features on the report run page ensure that reports answer even your toughest questions.



REPORT
Pipeline By Industry

Total Records 12 | Average Employees 3,070 | Total Amount $262,098.01

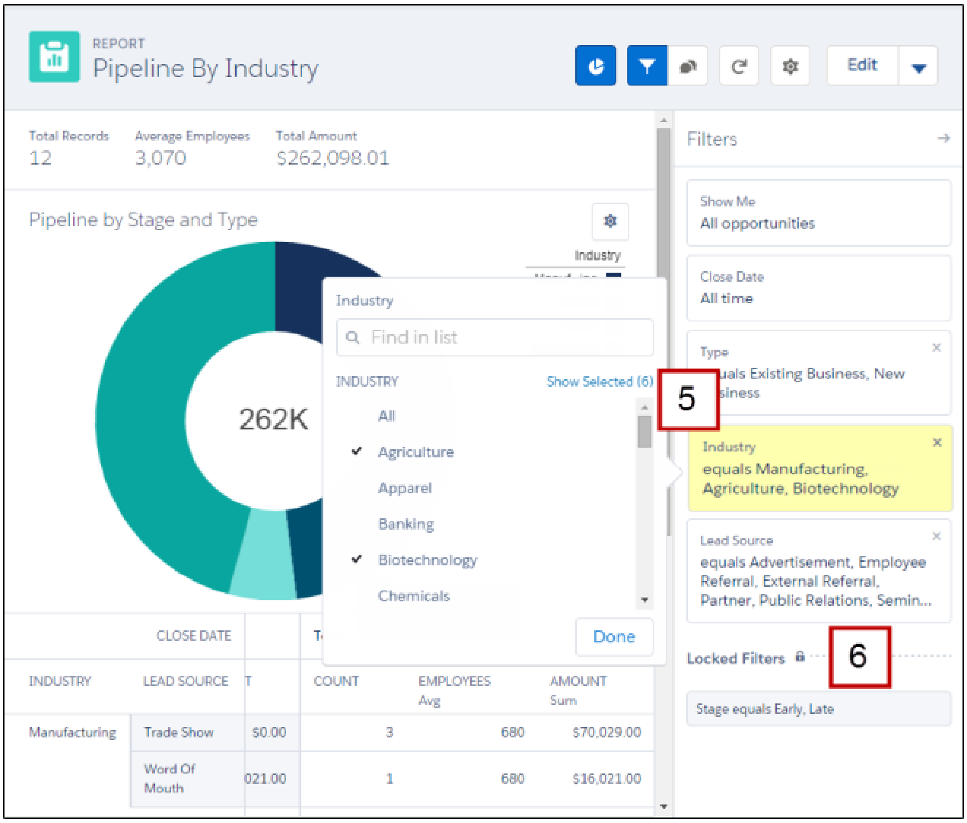
	CLOSE DATE	July 2015			Total		
INDUSTRY	LEAD SOURCE	COUNT	EMPLOYEES Avg	AMOUNT Sum	COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	0	0	$0.00	3	680	$70,029.00
	Word Of Mouth	1	680	$16,021.00	1	680	$16,021.00
	Subtotal: Manufacturing	1	680	$16,021.00	4	680	$86,050.00
Media	Partner	0	0	$0.00	2	14,668	$40,013.00
	Subtotal: Media	0	0	$0.00	2	14,668	$40,013.00
Retail	Word Of Mouth	1	0	$16,000.01	1	0	$16,000.01
	Subtotal: Retail	1	0	$16,000.01	1	0	$16,000.01
Technology	Advertisement	0	0	$0.00	2	0	$100,017.00
	Partner	0	0	$0.00	3	0	$20,018.00
	Subtotal: Technology	0	0	$0.00	5	0	$120,035.00
	Total	2	340	$32,021.01	12	3,070	$262,098.01

Get more from your report with the tools in the header (**1**).

- — Show or hide a report chart.
- — Add, remove, or change report filters. After applying a filter, the report automatically refreshes to show filtered data. You no longer have to open the report builder to filter a report.
- — Open the report feed to collaborate with others on report data.
- — Refresh your report to show the latest data.
- — Show or hide details like subtotals, grand totals, and record counts from your report.

Clicking **Edit** launches the report builder. The action menu provides one-click access to saving, cloning, and deleting the report.

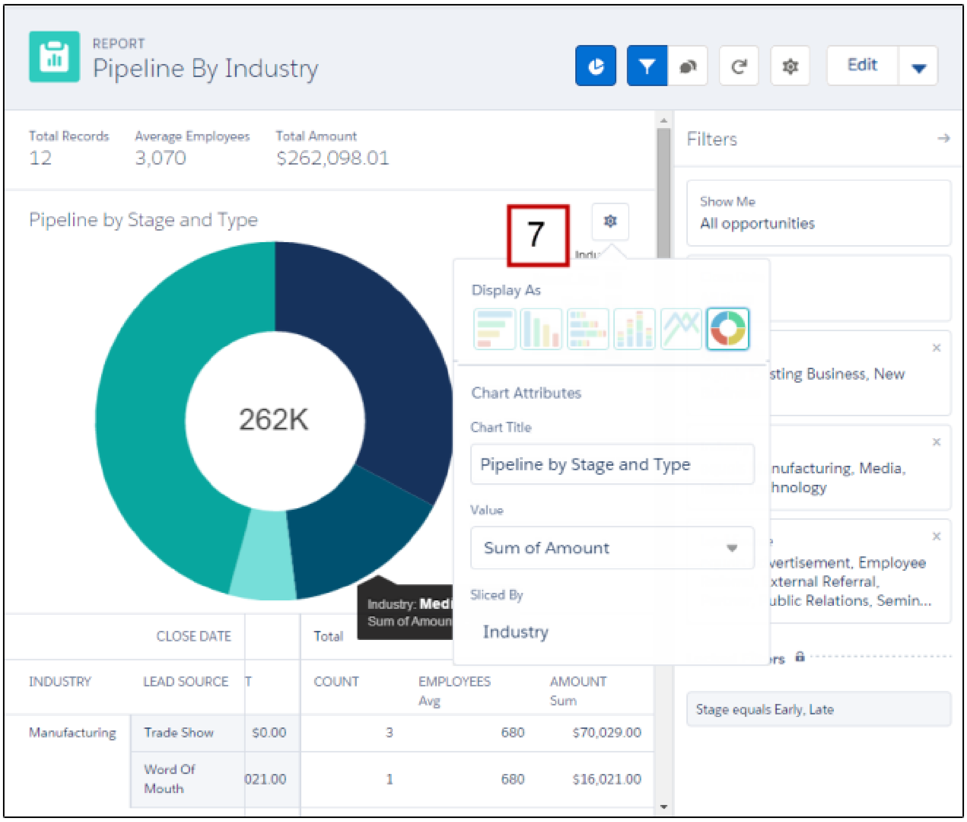
You can view key metrics at the top of every report (**2**). Report headers float on both the X and Y axes (**3**), so you always know what field you're reviewing without needing to scroll. And, we've redesigned the report format so that groupings are easier to read (**4**).



INDUSTRY	LEAD SOURCE		COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	$0.00	3	680	$70,029.00
	Word Of Mouth	021.00	1	680	$16,021.00

You can add, remove, and edit report filters (**5**) right from the report's page. You no longer have to open the report builder to filter the report you're reading. If you want to see your sales pipeline for the apparel industry, edit the `Industry` filter accordingly and your report refreshes.

You can also lock filters (**6**). If you want to share a report about your late-stage sales pipeline and don't want to share early-stage data, lock the `Stage` filter. Locked filters can't be edited on the report run page. You lock and unlock filters in the report builder.



Total Records	Average Employees	Total Amount
12	3,070	$262,098.01

Pipeline by Stage and Type

262K

Industry: Medi
Sum of Amoun

Chart Attributes

Chart Title: Pipeline by Stage and Type

Value: Sum of Amount

Sliced By: Industry

Show Me: All opportunities

sting Business, New

nufacturing, Media,
hnology

vertisement, Employee
xternal Referral,
ublic Relations, Semin...

Stage equals Early, Late

	CLOSE DATE		Total		
INDUSTRY	LEAD SOURCE	T	COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	$0.00	3	680	$70,029.00
	Word Of Mouth	021.00	1	680	$16,021.00

You're probably wondering: "But can I customize the chart?" Yes, you can (**7**)! Change the chart type, title, and more from the chart options menu (⚙).

Reports and Dashboards: Compatibility Between Lightning Experience and Salesforce Classic

Wondering how reports or dashboards created in one user interface work in the other? Here is a breakdown of what to expect.

For reports and dashboards created in Salesforce Classic:

- You can view and edit both in Lightning Experience.

- After saving a Salesforce Classic dashboard in Lightning Experience, you can't edit it in Salesforce Classic. Instead of editing a Salesforce Classic dashboard in Lightning Experience, consider cloning the dashboard and editing the clone. That way, you can still edit the original dashboard in Salesforce Classic.
- You can view and open folders that you created in Salesforce Classic in Lightning Experience. Lightning Experience obeys sharing rules set on report and dashboard folders in Salesforce Classic.
- Joined reports are available as a beta in Lightning Experience.

For reports and dashboards created in Lightning Experience:

- You can view and edit reports created in Lightning Experience in Salesforce Classic.
- You can't edit dashboards created in Lightning Experience in Salesforce Classic, but you can view them. Lightning tables, a table component only available in Lightning Experience, don't appear when you view a dashboard in Salesforce Classic.
- Dashboards that you create in Lightning Experience that have more than three columns automatically display in Salesforce Classic with three columns (retaining all dashboard components).

For more information about Lightning Experience reports and dashboards differences, see Reports and Dashboards: What's Different or Not Available in Lightning Experience in help.

CHAPTER 8 Your Layouts Can Customize Lightning Experience Records

In this chapter ...

You can customize the content of your record pages in Lightning Experience using tools you're already familiar with: page layouts and compact layouts.

Page Layouts in Lightning Experience

When you customize your page layouts in Salesforce Classic, those changes can 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 your org supports multiple page layouts, you can create a page layout 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.

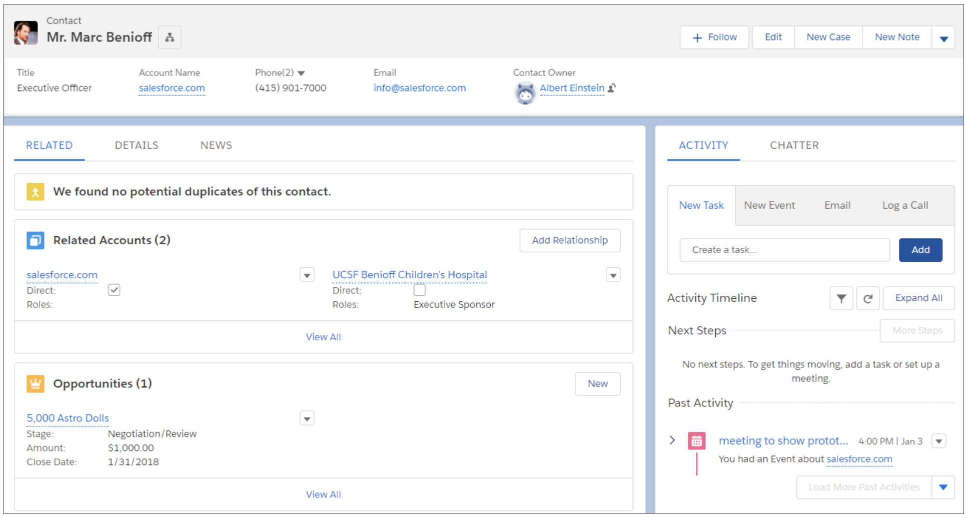
Here's a sample contact record in Lightning Experience. The highlights panel contains key record fields and is the only part of a record page that you can't customize using the page layout editor. The fields in the highlights panel are customized using a compact layout.

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



Contact
Mr. Marc Benioff

Title: Executive Officer
Account Name: salesforce.com
Phone(2): (415) 901-7000
Email: info@salesforce.com
Contact Owner: Albert Einstein

We found no potential duplicates of this contact.

Related Accounts (2)

salesforce.com
Direct: ✓
Roles:

UCSF Benioff Children's Hospital
Direct:
Roles: Executive Sponsor

Opportunities (1)

5,000 Astro Dolls
Stage: Negotiation/Review
Amount: $1,000.00
Close Date: 1/31/2018

Create a task...

Activity Timeline

Next Steps

No next steps. To get things moving, add a task or set up a meeting.

Past Activity

meeting to show protot... 4:00 PM | Jan 3
You had an Event about salesforce.com

These page layout elements are supported in Lightning Experience.

Actions

Actions display in different places, such as the highlights panel, Activity tab, and the Chatter tab. The actions are derived from the list of actions in the Salesforce Mobile 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 102.

Blank Spaces

Blank spaces are supported in Lightning Experience.

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.

The top-down tab-key order, which allows users viewing a record detail page to move through a column of fields from top to bottom before moving focus to the top of the next column of fields, isn't supported in Lightning Experience. Even if a page layout is configured for a top-down tab-key order, tabbing moves from left-to-right through field columns in Lightning Experience.

Related Lists

Related lists are included as Lightning components in Lightning Experience. Not all related lists are supported in Lightning Experience.

In Lightning Experience, the related list type determines how many fields are displayed in a related list. The **Basic List** related list type displays only the first four fields of a related list. The **Enhanced List** type shows up to 10 fields, lets you resize and sort columns, perform mass actions, and wrap text. To change the related list type, customize the related list component in the Lightning App Builder.

Report Charts

Report charts that you add to a page layout appear under the Details tab in Lightning Experience. When you add a report chart to a page layout, it can take a few moments before the chart appears on Lightning record pages.

Sections

Sections appear along with fields under the Details tab. A section with no header is incorporated into the section above it.

The Detail Page visibility setting controls whether the section header appears for both the detail page and the edit page. If the section header is set to display (or hide) on the detail page, the header also displays (or hides) on the edit page.

Standard and Custom Buttons

Standard and custom buttons are treated as actions in Lightning Experience, just like in the Salesforce mobile app.

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

Visualforce Pages

Visualforce pages that you added to the page layout appear under the Details tab. Only Visualforce pages with **Available for Lightning Experience, Lightning Communities, and the mobile app** enabled display in Lightning Experience on Lightning pages, utility bars, and the Salesforce mobile app.

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

- Expanded lookups
- Mobile cards
- S-controls
- Section header visibility for Edit Page
- Tags

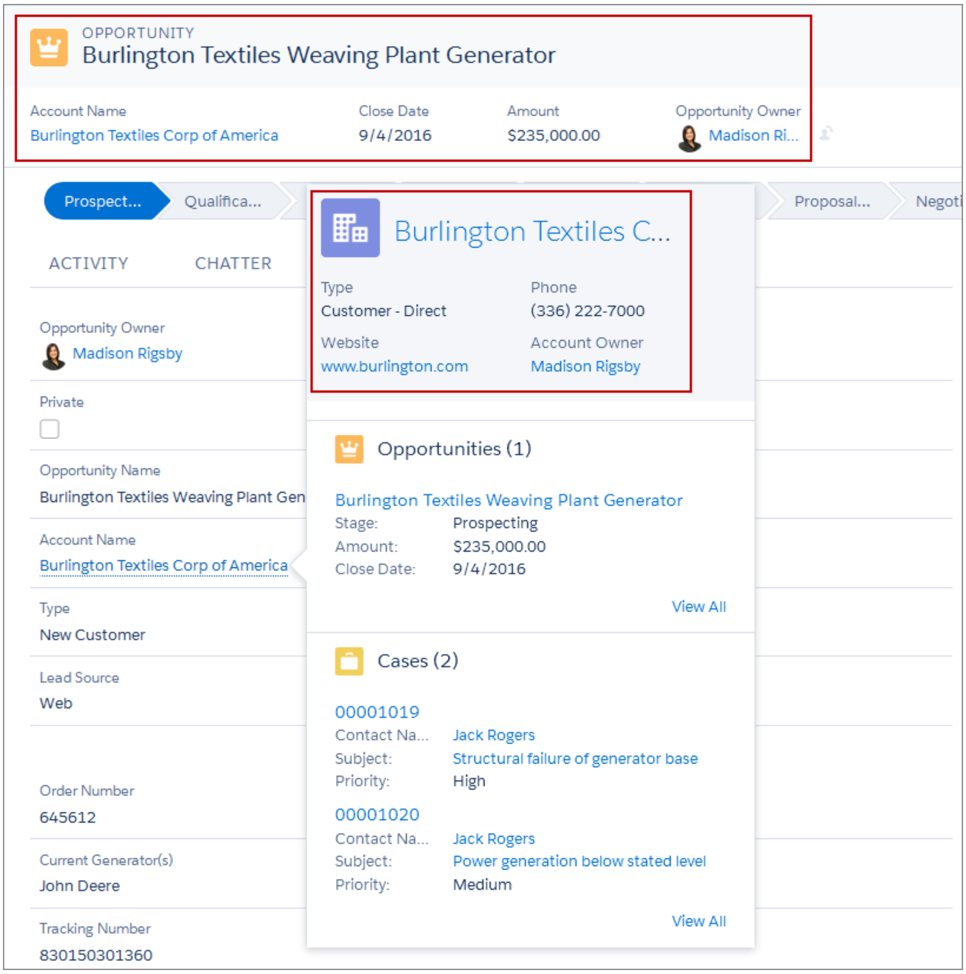
Note: You can't use the enhanced page layout editor to customize the layout of Lightning Experience record home pages.

Compact Layouts in Lightning Experience

If you've completed the Salesforce Mobile Basics module, you're familiar with compact layouts and how they work in the Salesforce mobile app. Compact layouts play the same role in Lightning Experience: displaying a record's key fields in the highlights panel of a record page. Compact layouts are also used in other places, such as the activity timeline and expanded lookup cards.

A compact layout lets you put the most important fields where your users can easily see them. If your org supports record types, you can assign a compact layout to different record types, just like you can with a page layout.

In Lightning Experience, up to the first seven fields in a compact layout appear in the highlights panel of an object record. (On smaller screens, the highlights panel displays fewer fields.) When a user hovers over a lookup relationship field on the object record page, a highlights panel for that field displays the first five fields from the compact layout. Highlights panels display the first field from the compact layout at the top in an accented font.


OPPORTUNITY
Burlington Textiles Weaving Plant Generator

Account Name: Burlington Textiles Corp of America
Close Date: 9/4/2016
Amount: $235,000.00
Opportunity Owner: Madison Ri...

Opportunity Owner: Madison Rigsby
Private: ☐
Opportunity Name: Burlington Textiles Weaving Plant Gen
Account Name: Burlington Textiles Corp of America
Type: New Customer
Lead Source: Web
Order Number: 645612
Current Generator(s): John Deere
Tracking Number: 830150301360

Burlington Textiles C...
Type: Customer - Direct
Phone: (336) 222-7000
Website: www.burlington.com
Account Owner: Madison Rigsby

Opportunities (1)
Burlington Textiles Weaving Plant Generator
Stage: Prospecting
Amount: $235,000.00
Close Date: 9/4/2016
View All

Cases (2)
00001019
Contact Na... Jack Rogers
Subject: Structural failure of generator base
Priority: High
00001020
Contact Na... Jack Rogers
Subject: Power generation below stated level
Priority: Medium
View All

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

Event and task compact layouts determine the fields that appear in the details section when you expand an activity in the activity timeline in Lightning Experience. When you change the compact layout for tasks in the activity timeline, you also impact the fields that show up in the highlights area on tasks, in tasks lists, and everywhere else the compact layout is used.

You can create and edit a compact layout from the Compact Layouts related list on any object in the Object Manager in Lightning Experience.

Changes you make to a compact layout are reflected in both Lightning Experience and the Salesforce mobile app.

CHAPTER 9 Actions and Lightning Experience

In this chapter ...

Actions enable users to do more in Salesforce, such as create or update records and log calls.

If you've already created and used actions in your organization, you're familiar with how they work in Salesforce Classic. If you've used our mobile apps, you've seen how they work in the Salesforce mobile app. In Lightning Experience, instead of showing up in one place—like the Chatter publisher or the Salesforce mobile app action bar—actions are split into different areas.

Next, we go over where you can find actions, which actions are and aren't supported, and how the customizations you've made to actions on a page layout affect how they display in Lightning Experience.

Actions in Lightning Experience

In Lightning Experience, actions display in the Global Actions menu in the header, on related lists, and on list view items. Actions also appear in several places on a record page. Where actions appear on a record page depends on the action's type.

Note: Quick actions aren't the only action type covered here. Standard and custom buttons are also considered actions.

EDITIONS

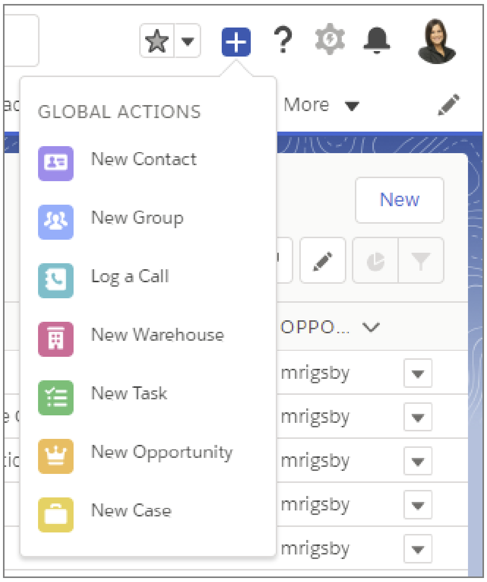
Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

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

Custom canvas actions available in: **Professional** (with 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 Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.



The items in the menu appear in the order that they're listed in the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.

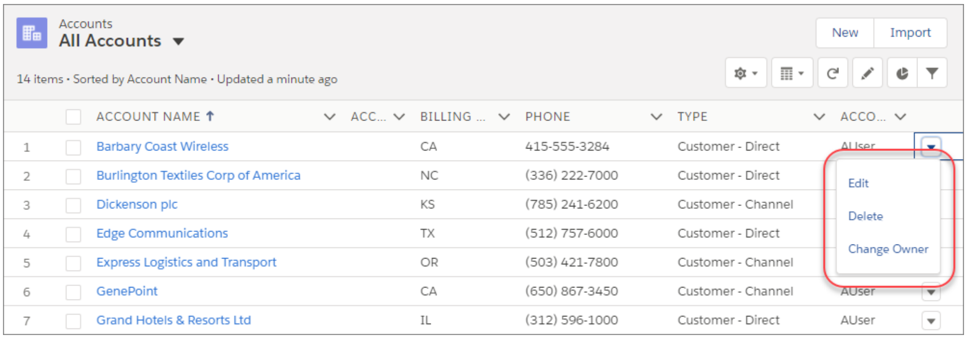
Actions associated with objects that aren't supported in Lightning Experience don't appear in the Global Actions menu. Also, the Global Actions menu doesn't support standard Chatter actions.

Actions on List Views and List View Items

Custom buttons, list view actions, and certain standard buttons are supported on all list views, except Recently Viewed. To have a custom button appear on a list view, add the button to the object's List View search layout.

In the Kanban view, only the standard New action is supported.

List view items support only specific standard actions, like Edit, Delete, or Change Owner.



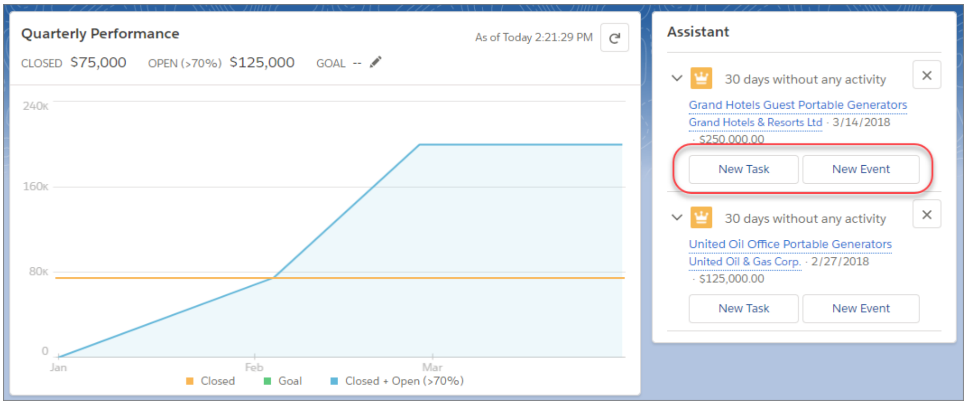
Accounts
All Accounts
New Import
14 items • Sorted by Account Name • Updated a minute ago

	ACCOUNT NAME	ACC...	BILLING ...	PHONE	TYPE	ACCO...
1	Barbary Coast Wireless		CA	415-555-3284	Customer - Direct	AUser
2	Burlington Textiles Corp of America		NC	(336) 222-7000	Customer - Direct	
3	Dickenson plc		KS	(785) 241-6200	Customer - Channel	
4	Edge Communications		TX	(512) 757-6000	Customer - Direct	
5	Express Logistics and Transport		OR	(503) 421-7800	Customer - Channel	
6	GenePoint		CA	(650) 867-3450	Customer - Channel	AUser
7	Grand Hotels & Resorts Ltd		IL	(312) 596-1000	Customer - Direct	AUser

For Tasks, table format list views and the Kanban view support only standard buttons. Tasks list view items in table view and the task item detail pane in split view contain the complete list of available actions for tasks.

Actions on the Home Page

On the Home page, you can find actions on recommendations in the Assistant. For example, imagine that 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.


Quarterly Performance
As of Today 2:21:29 PM
CLOSED $75,000 OPEN (>70%) $125,000 GOAL --
240K
160K
80K
0
Jan
Feb
Mar
Closed
Goal
Closed + Open (>70%)
Assistant
30 days without any activity
Grand Hotels Guest Portable Generators
Grand Hotels & Resorts Ltd · 3/14/2018 · $250,000.00
New Task
New Event
30 days without any activity
United Oil Office Portable Generators
United Oil & Gas Corp. · 2/27/2018 · $125,000.00
New Task
New Event

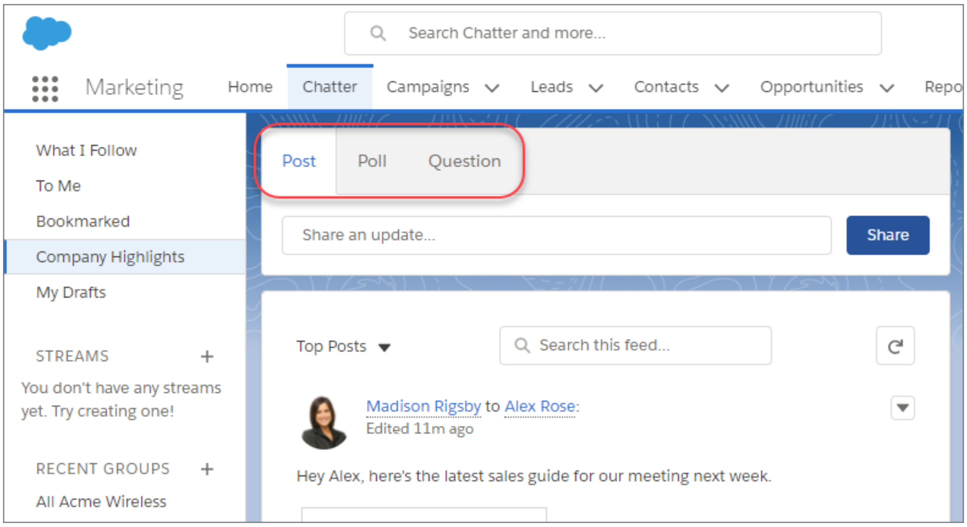
The actions that appear depend on the type of recommendation. To appear in the Assistant, actions must be added to the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout. Supported actions include:

- New Task
- New Event
- Edit
- Email

After you complete an action, the related recommendation disappears from the Assistant.

Actions on the Chatter Page

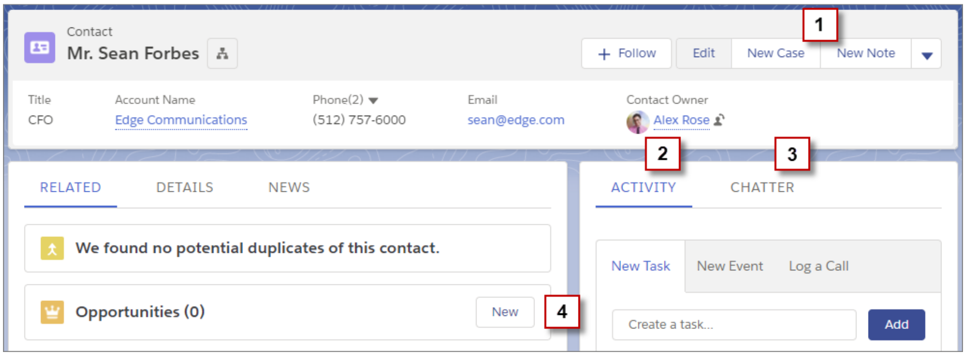
The Chatter page, like the Chatter tab on record pages, contains only standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. You can add, remove, or reorder the actions on the Chatter page from the Salesforce Mobile 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 actions related to creating tasks, creating events, and logging calls

- Standard buttons
- Custom object-specific Lightning component quick actions
- Custom flow actions
- Custom Visualforce quick actions
- Custom Visualforce buttons
- Canvas actions

The actions that appear in the page-level action menu are listed in the order that they appear in the Salesforce Mobile and Lightning Experience Actions section of the page layout.

The Activity tab (2) contains Create a Record quick actions that point to the Event and Task objects. It also contains Log A Call and Send Email actions.

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.

Note: Actions on user profiles, cases, and work orders can appear in a different way than on other records.

- Actions on the user profile page come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout.
- When feed tracking is enabled for cases or work orders, the page-level action menu on those records contains only custom buttons and supported standard buttons. Quick actions appear on the Chatter tab.

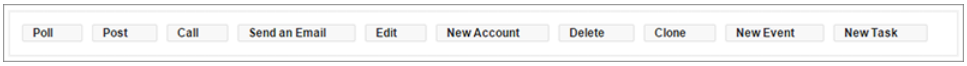
Actions on Related Lists

Related lists in Lightning Experience (4) show custom list buttons and supported standard buttons assigned to the related list. Not all related list standard buttons are supported in Lightning Experience.

Actions on Reports

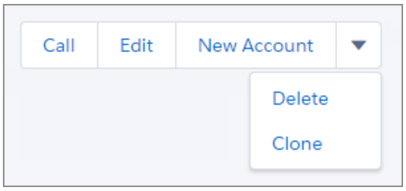
Actions on reports come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. However, only standard Chatter actions appear on reports, regardless of which other actions are assigned to the global publisher layout.

Example: Let's say you have these actions on your Contact page layout in the Salesforce Mobile and Lightning Experience Actions section.

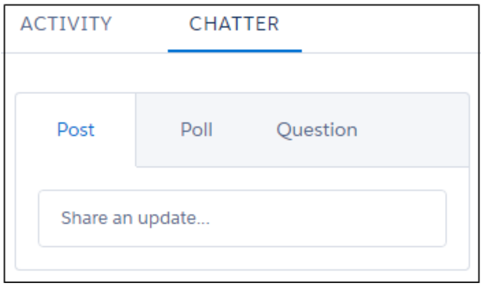


You have quick actions (New Account, New Event, New Task), a productivity action (Call), standard buttons (Edit, Delete, Clone, Send an Email), and Chatter actions (Poll, Post). Here's how those actions appear on a contact record page in Lightning Experience.

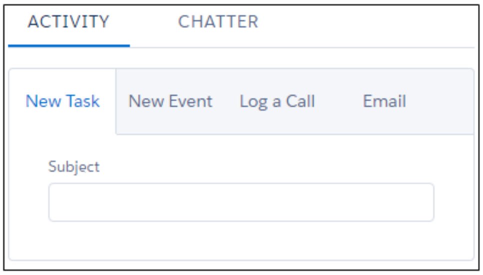
- The actions in the page-level action menu are a combination of the quick actions, productivity actions, and standard buttons. These actions appear in the order that they're listed on the page layout. Although they're quick actions, New Event and New Task don't show up here.



- 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.



How Actions Are Ordered in Lightning Experience

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) 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 Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce Mobile 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 Salesforce Mobile 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

However, the order of the actions from the global page layout aren't respected on those record pages.

When you click to override the predefined actions in the Salesforce Mobile and Lightning Experience Actions section, the custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the custom buttons as actions from the Mobile & Lightning Actions category in the palette.

After you customize the Salesforce Mobile and Lightning Experience Actions section of an object's page layout, the actions in each section of the record page respect the ordering of its types of actions on the page layout. For example, actions in the page-level actions menu appear in the order that you configure them in the Salesforce Mobile and Lightning Experience Actions section on the page layout. And actions in the Chatter and Activity tabs reflect the order of the actions supported for those tabs on the page layout.

The Global Actions menu (➕) in the Lightning Experience header displays all global quick actions from the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout, except the standard Chatter actions Post, File, Poll, Link, Question, and Thanks.

ADJUST YOUR EXISTING CUSTOMIZATIONS FOR LIGHTNING EXPERIENCE

CHAPTER 10 Get Started with the Lightning Experience Configuration Converter

In this chapter ...

The Lightning Experience Configuration Converter is a standalone tool that lives outside of Salesforce. It helps you prepare your org for Lightning Experience by streamlining or automating common transition tasks.

EDITIONS

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

Available in: English language only

Note: Get the full picture of your org by running the Lightning Experience Readiness Check as well as using the Lightning Experience Configuration Converter.

Note: The Lightning Experience Configuration Converter is a standalone application that lives outside your ordinary Salesforce environment. When you use it, your account will be used to pull relevant data from your Salesforce org into Heroku, where the data will be processed to provide the Configuration Converter's functionality. Heroku is another service offered by Salesforce on a separate infrastructure. The Configuration Converter may also use further downstream processors, including third parties. Tell me more about Heroku.

Scan Your Org for Updates with the Lightning Experience Configuration Converter

The Lightning Experience Configuration Converter searches your org for customizations that may need adjusting and speeds up getting them ready for the new interface. Use it to evaluate and update items such as Visualforce pages, JavaScript buttons and links, hard-coded URLs, actions and buttons, and AppExchange packages.

Note: Get the full picture of your org by running the Lightning Experience Readiness Check as well as using the Lightning Experience Configuration Converter.

Note: The Lightning Experience Configuration Converter is a standalone application that lives outside your ordinary Salesforce environment. When you use it, your account will be used to pull relevant data from your Salesforce org into Heroku, where the data will be processed to provide the Configuration Converter's functionality. Heroku is another service offered by Salesforce on a separate infrastructure. The Configuration Converter may also use further downstream processors, including third parties. Tell me more about Heroku.

To check your progress, Salesforce recommends scanning each feature area at least one time each release.

EDITIONS

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

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:
- Customize Application, Modify All Data

To convert JavaScript buttons and links or update hard-coded URLs:
- My Domain is deployed in your org

1. Open a tab in your browser, and log in to the Lightning Experience Configuration Converter at https://lightning-configuration.salesforce.com/.

   We recommend running the tool in a sandbox or Developer Edition org, and then migrating changes to your production org.

2. Review the status of each feature on the Home tab.
3. To update the status of a feature, kick off a scan of that feature from the Home tab.

   Note: While the Configuration Converter is running a scan, you can’t view scan results on any tab, take conversion actions, or kick off another scan. You can close the Configuration Converter while a scan is in progress. Salesforce sends you an email when the scan is finished.

4. If this is your first time using the tool, scan each feature.

5. Review and act on scan results.

Review and Act on Lightning Experience Configuration Converter Scan Results

EDITIONS

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

Available in: English language only

Use Lightning Experience Configuration Converter scan results to target conversions that make the most impact on the users you're transitioning next. In many cases, the Configuration Converter can convert features at the click of a button.

Select the tab for a feature to see detailed scan results. Scan results include guidance and recommendations on updating potential issues. In many cases, the Lightning Experience Configuration Converter lets you automate the recommended updates. For others, it walks you through manual updates.

- Actions and Buttons
- JavaScript Buttons and Links
- Hard-Coded URLs
- Visualforce Pages
- AppExchange Packages

Before taking recommended actions, use the current data to verify that your rollout plan is focused on the correct priorities for the group of users you're transitioning next. We recommend using the **User Impact** column to help focus on just the items needed for relevant users. Review Discover Phase: Start Your Lightning Experience Transition for help with planning your rollout and change management strategies.

Lightning Experience Configuration Converter Considerations

Keep these considerations in mind when working with the Configuration Converter.

General Considerations

- The Configuration Converter is a standalone tool that lives outside of Salesforce. Whitelist these IP addresses for your organization so admins can access the tool.
  - 52.2.2.209

  - 52.203.195.34
  - 52.200.207.131
  - 52.72.119.132

- The tool scans these objects. The JavaScript Buttons tab includes buttons from additional objects not listed here, but you can't convert them using the tool.
  - Account
  - Asset
  - Campaign
  - Case
  - Contact
  - Contract (JavaScript Buttons tab only)
  - Event
  - Lead
  - Opportunity
  - Order
  - Person Account
  - Task (JavaScript Buttons tab only)
  - Custom objects
- If a page layout name contains special characters, some actions to that layout won't work using the tool.

Considerations for JavaScript Buttons and Links

- The converter's JavaScript Buttons area suggests Lightning components for some conversions. As a result, you must have My Domain deployed in your org to convert JavaScript buttons.
- Not all JavaScript buttons and links can be recreated. For example, the tool can't recreate some items that contain a URL that sets predefined values (also known as a URL hack).
- Some complex JavaScript alerts, such as those with multiple operations, must be converted manually. The converter recommends actions that you can take to perform the manual conversion.
- When you deploy a converted quick action to a page layout whose Salesforce Mobile and Lightning Experience Actions section isn't customized, the tool overrides the defaults in that section and then deploys the action.

- If a JavaScript custom button or link isn't used on any page layouts, you can convert it but not deploy it. The converted item is added to the palette of the object's page layouts, and you can then add it to a page layout manually.
- List view buttons and items from managed packages aren't supported for conversion. To get a list of your managed packages, run the Lightning Experience Readiness Report and look under AppExchange Packages.
- If your JavaScript button or link code uses REQUIRESCRIPT, the only scripts that are supported for conversion are:
  - `/soap/ajax/xx.x/connection.js`
  - `/soap/ajax/xx.x/apex.js`

Considerations for Actions and Buttons

- For the Case object, Change Status and Log a Call actions aren't moved because the Salesforce Mobile and Lightning Experience Actions section doesn't support them. Make the necessary updates manually.
- If you don't see a step in the wizard, such as selecting a record type or removing the Mobile Smart Actions bundle, it's not applicable to the selected page layouts.
- If a page layout's Quick Actions in the Salesforce Classic Publisher section hasn't been customized, you can't select the page layout.
- Actions and buttons from objects in managed packages aren't supported for moving.

Considerations for Hard-Coded URLs

- The Configuration Converter scans the following for hard-coded URLs.
  - Custom buttons or links
  - Visualforce pages
  - Visualforce components
  - Apex triggers
  - Apex classes
  - Collaboration groups
  - Email templates
  - Knowledge articles
  - Home Page content
  - Web tabs

  - Document URLs

- Hard-coded URLs in Chatter posts and Knowledge articles are displayed in the scan results, but they can't be updated by the Configuration Converter.
- Hard-coded URLs in email templates and document URLs can't be updated if they are stored in the user's personal folder or in the general public folder.
- URLs in packages and workflows aren't scanned and can't be updated by the Configuration Converter.
- The Configuration Converter displays each hard-coded URL as a separate line item. Previously, the Readiness Check grouped duplicate URLs into a single row.

CHAPTER 11 Move Actions and Buttons to Lightning Experience

The Lightning Experience Configuration Converter scans your org for actions and buttons. You can move all the actions and buttons at once, or let Salesforce guide you through the move process for each object.

EDITIONS

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

Available in: English language only

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:

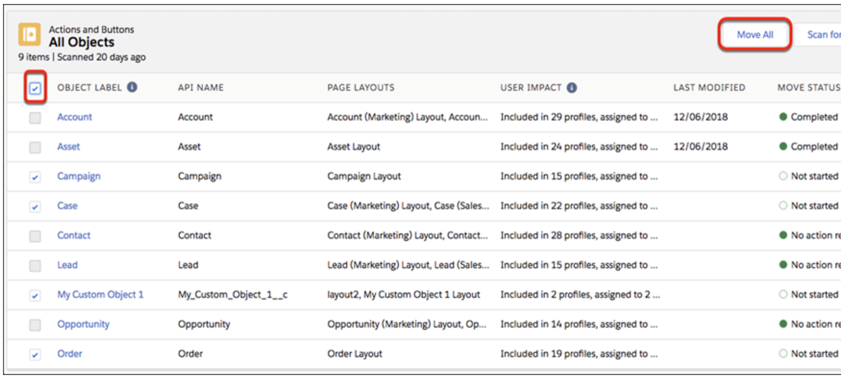
- Customize Application, Modify All Data

To convert JavaScript buttons and links or update hard-coded URLs:

- My Domain is deployed in your org

Opt to move your actions and buttons all at once, or one at a time.

1. Log in to the Lightning Experience Configuration Converter, then select the **Actions and Buttons** tab.
2. If your org hasn't been scanned for actions and buttons before, or if the scan data is out of date, scan for actions and buttons to get the latest data.
3. Select all actions and buttons by clicking the select box in the header. Or, pick which actions and buttons to move by selecting individual rows. You can always move the rest later.
4. Click **Move Selected**.



Actions and Buttons
All Objects
9 items | Scanned 20 days ago

✓	OBJECT LABEL	API NAME	PAGE LAYOUTS	USER IMPACT	LAST MODIFIED	MOVE STATUS
	Account	Account	Account (Marketing) Layout, Accoun...	Included in 29 profiles, assigned to ...	12/06/2018	● Completed
	Asset	Asset	Asset Layout	Included in 24 profiles, assigned to ...	12/06/2018	● Completed
✓	Campaign	Campaign	Campaign Layout	Included in 15 profiles, assigned to ...		○ Not started
✓	Case	Case	Case (Marketing) Layout, Case (Sales...	Included in 22 profiles, assigned to ...		○ Not started
	Contact	Contact	Contact (Marketing) Layout, Contact...	Included in 28 profiles, assigned to ...		● No action r
	Lead	Lead	Lead (Marketing) Layout, Lead (Sales...	Included in 15 profiles, assigned to ...		● No action r
✓	My Custom Object 1	My_Custom_Object_1__c	layout2, My Custom Object 1 Layout	Included in 2 profiles, assigned to 2 ...		○ Not started
	Opportunity	Opportunity	Opportunity (Marketing) Layout, Op...	Included in 14 profiles, assigned to ...		● No action r
✓	Order	Order	Order Layout	Included in 19 profiles, assigned to ...		○ Not started

5. A confirmation window appears that summarizes the steps we'll take once you confirm. While the move takes place, you can't work in the other tabs. Salesforce emails you when the move is complete.

CHAPTER 12 Convert JavaScript Buttons and Links into Lightning Experience Alternatives

In this chapter ...

Uncomfortable with code? Stymied by JavaScript? Don't worry. With just a few clicks in the Lightning Experience Configuration Converter, you can recreate many JavaScript buttons and links as Lightning components, quick actions, or other declarative solutions.

The Lightning Experience Configuration Converter creates new Lightning components, quick actions, and other declarative solutions without touching the original items. Before committing to any changes, you can preview the new component code or the declarative steps to verify that the alternatives work.

Note: Starting in October 2019, JavaScript Buttons and Links scan results are only available in the Lightning Experience Configuration Converter. Admins do not receive scan results as email attachments.

EDITIONS

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

Available in: English language only

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:

- Customize Application, Modify All Data

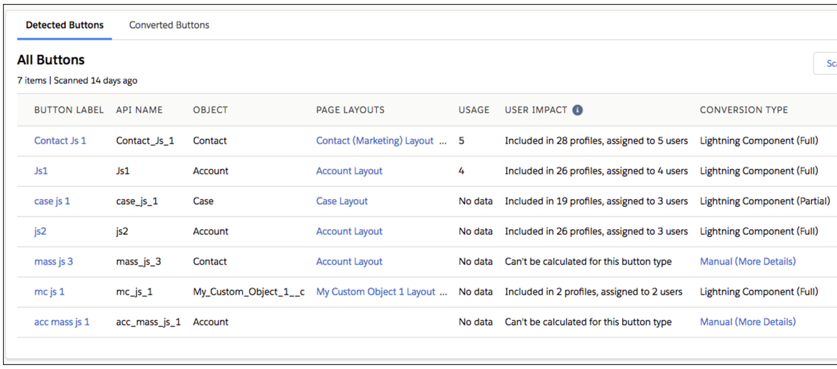
To create, edit, or delete Visualforce pages:

- Customize Application

1. Log in to the Lightning Experience Configuration Converter, then select the **JavaScript Buttons** tab.
2. If no data exists or if the existing scan data is outdated, click Scan for JavaScript Buttons to see the latest information. The last scanned date is displayed in the table header.

3. When the scan finishes, all the JavaScript buttons and links found are listed for each object. You can see which page layout each item is on and the recommended conversion type.

4. To see conversion options for an item, click ▼ on its row.



All Buttons

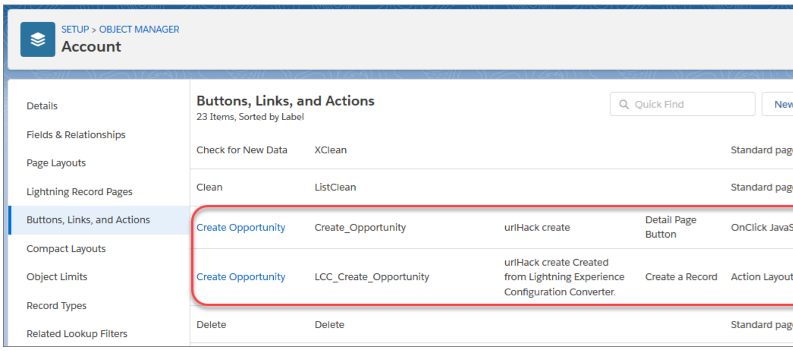
7 items | Scanned 14 days ago

BUTTON LABEL	API NAME	OBJECT	PAGE LAYOUTS	USAGE	USER IMPACT	CONVERSION TYPE
Contact Js 1	Contact_Js_1	Contact	Contact (Marketing) Layout ...	5	Included in 28 profiles, assigned to 5 users	Lightning Component (Full)
Js1	Js1	Account	Account Layout	4	Included in 26 profiles, assigned to 4 users	Lightning Component (Full)
case js 1	case_js_1	Case	Case Layout	No data	Included in 19 profiles, assigned to 3 users	Lightning Component (Partial)
js2	js2	Account	Account Layout	No data	Included in 26 profiles, assigned to 3 users	Lightning Component (Full)
mass js 3	mass_js_3	Contact	Account Layout	No data	Can't be calculated for this button type	Manual (More Details)
mc js 1	mc_js_1	My_Custom_Object_1__c	My Custom Object 1 Layout ...	No data	Included in 2 profiles, assigned to 2 users	Lightning Component (Full)
acc mass js 1	acc_mass_js_1	Account		No data	Can't be calculated for this button type	Manual (More Details)

all items, except the ones marked as `Manual (More Details)` or `not supported`, you can:

- Preview—View what the suggested alternative is, whether it's declarative or programmatic.
- Convert—Convert the item into the suggested alternative, but don't add it to the page layout.
- Convert & Deploy—Convert the item into the suggested alternative, and add it to the page layout. Quick actions, Lightning components, and custom buttons are added as actions to the Salesforce Mobile and Lightning Experience Actions section of the page layout. Converted links are added to the Custom Links area of the page layout's detail section.

5. If you want to convert the item but not deploy it to your users yet, click ▼ , and select **Convert**.
The prefix "LCC" is added to the converted item's API Name.



Buttons, Links, and Actions
23 Items, Sorted by Label

Check for New Data	XClean			Standard pag
Clean	ListClean			Standard pag
Create Opportunity	Create_Opportunity	urlHack create	Detail Page Button	OnClick JavaS
Create Opportunity	LCC_Create_Opportunity	urlHack create Created from Lightning Experience Configuration Converter.	Create a Record	Action Layout
Delete	Delete			Standard pag

Because the item was converted but not deployed, it wasn't added to the page layout.

When you're ready for your users to use the converted item, you can deploy it in two ways.

- Go to the affected page layout in Setup. In the palette, locate the new item under Mobile & Lightning Actions and manually add it.
- In the Lightning Experience Configuration Converter, on the Converted Buttons tab, find the converted item, click ▾ , and select **Deploy**.

6. To convert an item and add it to the page layout, click ▾ , and select **Convert & Deploy**.

   Important: For partial conversions, the component code is incomplete. We recommend that you convert it but not use the tool to deploy it. If you deploy the incomplete code, the item doesn't work properly.

7. After you deploy the converted button or link, we recommend that you remove the original one from the page layout. You can mouse over the items in the palette to see which one is the new converted item. When you're satisfied with the replacement, consider deleting the original JavaScript item from your org.

JavaScript Button and Link Conversion and Deployment Behavior

The Lightning Experience Configuration Converter recognizes several types of JavaScript buttons and links, including alerts, redirects, urlHacks, popups, and confirmation dialogs. Depending on what the button or link does, the tool suggests a declarative or programmatic alternative that's compatible with Lightning Experience.

In all conversions, your original JavaScript button or link remains untouched.

EDITIONS

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

Conversion Type	Description	Conversion and Deployment Behavior
Custom Button or Link	You can convert the JavaScript button or link to a regular custom button or link.	Upon conversion, a button or link matching the functionality and label of the JavaScript button or link is added to your org on the same object as the original. For example, suppose that this org has a JavaScript detail page link called "Visit Website." After conversion, a custom URL link with the same name appears in the Buttons, Links, and Actions list for the object. The prefix "LCC" is added to the converted button or link's API Name. When deployed, custom buttons are added as actions to the Salesforce Mobile and Lightning Experience Actions section of the affected page layout. Deployed links are added to the Custom Links area of the page layout's detail section.
Quick Action	You can convert the JavaScript button to a quick action.	Upon conversion, a quick action matching the functionality and label of the JavaScript button is added to your org on the same object as the original. For example, suppose that this org has a JavaScript button called "Create Opportunity." After conversion, a quick action with the same name appears in the Buttons, Links, and Actions list for the object. The prefix "LCC" is added to the converted button or link's API Name.

Conversion Type	Description	Conversion and Deployment Behavior
		When deployed, the quick action is added to the Salesforce Mobile and Lightning Experience Actions section of the affected page layout.
Lightning Component (Full)	You can convert the JavaScript button or alert to a Lightning component quick action and retain its full functionality.	Upon conversion, a Lightning component bundle with component code and a controller file is added to your org. You can find it in Setup under Custom Code | Lightning Components. In addition, a Lightning component action that triggers the new component is created. The Lightning component action has the same name as the original JavaScript button or link. When deployed, the Lightning component action is added to the Salesforce Mobile and Lightning Experience Actions section of the affected page layout.
Lightning Component (Partial)	You can partially convert the JavaScript button or alert to a Lightning component.	Upon conversion, a Lightning component bundle with component code and a controller file is added to your org. A Lightning component action that triggers the new component is also created. The Lightning component action has the same name as the original JavaScript button or link. Complete the component before deploying it. If you don't have access to a developer to help you with the code, post your issue on the Lightning Experience Configuration Converter Trailblazer Community. Folks there would be glad to help. After the code has been finished to match the functionality of the JavaScript button or link, manually add its Lightning component quick action to the page layout.

Conversion Type	Description	Conversion and Deployment Behavior
Manual (More Details)	The Lightning Experience Configuration Converter can't convert the item.	Click **Manual (More Details)** to get more information and suggested replacements. Sometimes, the reason is because the type isn't supported for conversion, such as a list view or related list button. In other cases, the JavaScript button or alert is too complex to convert declaratively.

You can view the converted items in the Buttons, Links, and Actions list in the Object Manager in Setup.

When the Lightning Experience Configuration Converter Can't Convert Your JavaScript Buttons or Links

EDITIONS

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

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:

- Customize Application, Modify All Data

To convert JavaScript buttons and links or update hard-coded URLs:

- My Domain is deployed in your org

Sometimes the Lightning Experience Configuration Converter finds a JavaScript button or link that's either not a supported type—like a related list or list view button— or is too complex to convert. When that happens, you see "Manual (More Details)" in the Conversion Type column. Here's what to do next.

When you click **Manual (More Details)** and the button isn't supported, you get a message that you can't covert the item.

If your button or link matches one of the common archetypes, such as a dialog or urlHack, the message recommends one or more components in the Sample Components for Lightning Component Actions repo on GitHub. This repo contains sample Lightning component bundles that represent some of the most common JavaScript button use cases. They've been created with extra annotation to help you configure them to your own needs.

You don't need a GitHub login to access the repo, and you can install the components in your org using an unmanaged package.

Tip: Admins, you can do this! Installing the Lightning components from GitHub takes only a few clicks.

1. Determine what your comfort level with code is.

- If the thought of working with code makes you break out in a cold sweat, and you don't have anyone to help you, fear not! We're not going to abandon you. First, note which components the converter recommended. Install the components in your org. Then post your issue on the Lightning Experience Configuration Converter Trailblazer Community. The folks there would be glad to help you.
- If you're not comfortable with code, but you have a developer that can help you, install the component bundle unmanaged package from GitHub. You can then work through it together.
- If you're comfortable with code, go to the GitHub repo, and look at the available components. Scroll through the Classes & Components list, and see which of the recommended components is similar to what your JavaScript button does. Install the component bundle unmanaged package in your org.

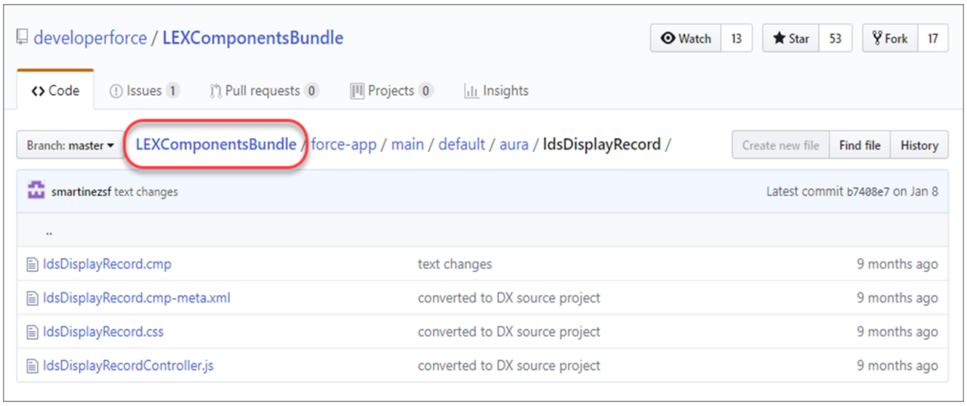
Ready to install the components? Let's go.

2. Make sure that you're logged in to the Salesforce sandbox or Developer Edition org that you want to use to work on the button conversion.

   We recommend not installing these components in your production org.

3. Go to the home page for the component bundle on GitHub.

   - In the Lightning Experience Configuration Converter, clicking a link in the Manual (More Details) dialog brings you directly into a component file in GitHub. Navigate to the top level by clicking **LEXComponentsBundle** in the directory structure breadcrumb.

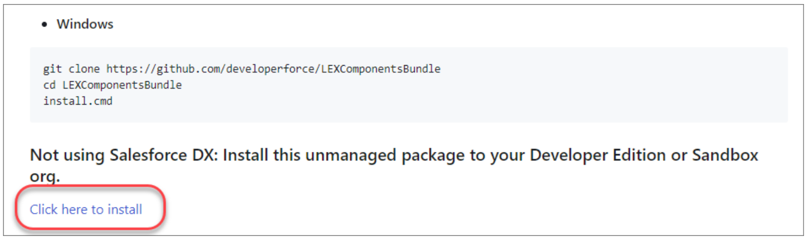

developerforce / LEXComponentsBundle
Watch 13
Star 53
Fork 17
Code
Issues 1
Pull requests 0
Projects 0
Insights
Branch: master
LEXComponentsBundle / force-app / main / default / aura / IdsDisplayRecord /
Create new file
Find file
History

smartinezsf text changes — Latest commit b7408e7 on Jan 8

..		
IdsDisplayRecord.cmp	text changes	9 months ago
IdsDisplayRecord.cmp-meta.xml	converted to DX source project	9 months ago
IdsDisplayRecord.css	converted to DX source project	9 months ago
IdsDisplayRecordController.js	converted to DX source project	9 months ago

   - Otherwise, open a new browser tab, and go to https://github.com/developerforce/LEXComponentsBundle.

4. Scroll down the page until you find the **Click here to install** link, and click it.



- Windows

```
git clone https://github.com/developerforce/LEXComponentsBundle
cd LEXComponentsBundle
install.cmd
```

Not using Salesforce DX: Install this unmanaged package to your Developer Edition or Sandbox org.

Click here to install

5. If you're asked, log in with your org credentials.
6. Leave the settings as they are, and click **Install**.

   The installation can take a few moments.

7. When the installation finishes, click **Done**.
8. To find the installed components, in Setup, enter *`Components`* in the Quick Find box, then click **Lightning Components**.

   Not going to customize the code yourself? Get help to make it happen, and skip to the last step. Otherwise, read on.

9. Click the name of the component you want to configure, and click **Developer Console**.
10. Update the component code to match the functionality of the JavaScript button or link that it's replacing.
11. When the updated Lightning component is ready to use, create a Lightning component quick action to trigger it.
12. Go to the affected page layout in Setup. Find the Lightning component quick action that you created in the palette, and add it to the Salesforce Mobile and Lightning Experience Actions section of the page layout.

    You can find Lightning component actions in the palette under Mobile & Lightning Actions.

CHAPTER 13 Update References to Hard-Coded URLs for Lightning Experience

Some hard-coded references to your org's original URL won't work after you enable My Domain. Use the Hard-Coded URLs' tab in the Lightning Experience Configuration Converter to locate your hard-coded URLs, and then update them with a single click.

Opt to move your hard-coded URLs all at once, or one at a time.

1. Log in to the Lightning Experience Configuration Converter, then select the **Hard-Coded URLs** tab.
2. If your org hasn't been scanned for hard-coded URLs before, or if the scan data is out of date, scan to get the latest data.
3. Select all hard-coded URLs by clicking the select box in the header. Or, pick which URLs to move by selecting individual rows. You can always move the rest later.
4. Click **Update Selected URLs**.
5. A confirmation window appears that summarizes the steps we take after you confirm. While the update takes place, you can't work in the other tabs. Salesforce emails you when updates are completed.

EDITIONS

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

Available in: English language only

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:

- Customize Application, Modify All Data

To convert JavaScript buttons and links or update hard-coded URLs:

- My Domain is deployed in your org

CHAPTER 14 Prepare Your Visualforce Pages for Lightning Experience

In this chapter ...

Scan your Visualforce pages to identify issues that may affect your transition to Lightning Experience. Visualforce pages are grouped by profile and prioritized by greatest user impact. Apply the Lightning look and feel to one or more Visualforce pages automatically with Lightning Experience Stylesheets, no code required. Always test your Visualforce page in Lightning Experience first—most work as-is. If the listed issue prevents the page from working properly, follow the instructions to tweak the markup. If a Visualforce page requires significant refactoring, move it to Lightning Components.

EDITIONS

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

Note: Starting in October 2019, Visualforce feature scan results are only available in the Lightning Experience Configuration Converter. Admins do not receive scan results as email attachments.

Apply Lightning Stylesheets to Visualforce Pages Automatically

EDITIONS

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

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:

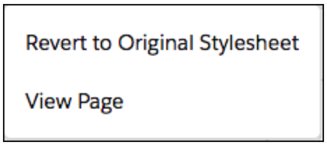
- Customize Application, Modify All Data

To create, edit, or delete Visualforce pages:

- Customize Application

The Lightning Experience Configuration Converter can apply Lightning Experience Stylesheets to one or more Visualforce pages automatically, no code required.

1. Log in to the Lightning Experience Configuration Converter, then select the **Visualforce Pages** tab.
2. If your org hasn't been scanned for Visualforce pages before, or if the scan data is out of date, scan to get the latest data.
3. Select all Visualforce pages by clicking the select box in the header. Or, pick which Visualforce pages to apply stylesheets to by selecting individual rows.
4. Click **Apply Lightning Experience Stylesheets**.
5. To change a Visualforce page back to its original style sheet, select **Revert to Original Stylesheet** from the dropdown menu. You can only revert one Visualforce page to its original style sheet at a time.



Update Visualforce Markup for Lightning Experience

EDITIONS

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

Some Visualforce pages may need tweaking to work in Lightning Experience. Don't worry if you don't have Visualforce experience—the workarounds show you exactly what to do each step of the way. Always test your Visualforce page in Lightning Experience first—most work as-is. If the listed issue is preventing the page from working properly, follow the instructions to tweak the markup and make it work. If a Visualforce page requires significant refactoring, move it to Lightning Components.

These workarounds are meant to guide you in the right direction. If you're unable to find a working solution, post questions to the Lightning Exp Configuration Converter group in the Trailblazer Community. Use the #VisualforceInLightning hashtag to call on our experts.

Reconfigure iframe Components

When a URL is displayed in an inline frame (iframe) on a Visualforce page, it can cause errors in Lightning Experience. Sometimes your URL looks different than you want, or maybe the whole page doesn't load in Lightning Experience. Most iframes are compatible with Visualforce in Lightning Experience. However, if you preview your Visualforce page in Lightning Experience and don't like what you see, you can usually fix it.

EDITIONS

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

USER PERMISSIONS

To create, edit, or delete Visualforce pages:
- Customize Application

To create and save Lightning pages in the Lightning App Builder:
- Customize Application

To create and edit custom tabs:
- Customize Application

1. Locate the iframe in the Visualforce code.
   a. From Setup, enter `Visualforce` in the **Quick Find** box, then select **Visualforce Pages**.
   b. Click **Edit** next to the Visualforce page.
   c. Search the Visualforce markup for each of the following: `apex:iframe`, `iframe`, and `apex:canvasApp`.

   Note: Address each instance that you find separately.

2. If you find any instances of `apex:canvasApp`, explore these other options to make it available in the UI.
3. If you find any instances of `apex:iframe` or `iframe`, find the source of the iframe.
   a. The source can be found after the `src=`. If you see `$Page`, the iframe is referencing another Visualforce page. Otherwise, the source is a URL.

```
<apex:iframe src="{!$Page.AccountListView}"/>

<iframe src="http://www.salesforce.com"/>
```

4. If the source of the iframe is a URL, use a custom web tab instead of the iframe.
   a. Identify the URL. View the Visualforce markup, and find the URL that follows src=.
   b. Remove the reference to the iframe from the markup. Be sure to copy the source URL.

```
<apex:iframe src="https://www.salesforce.com"/>
```

c. Save your changes.

d. Follow these instructions, using the same URL that you copied from the iframe in the Visualforce page.

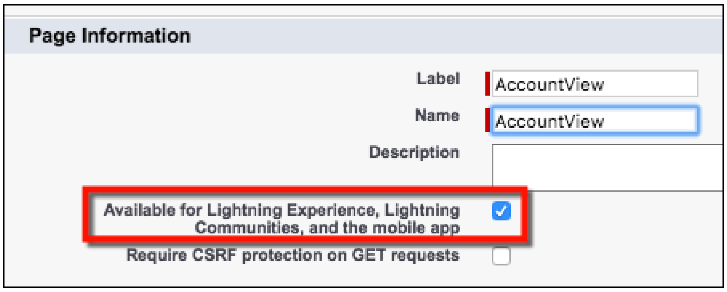
5. If the source of the iframe is another Visualforce page, create a Lightning Record Page to display both Visualforce pages separately.

   a. Identify the Visualforce page. View the Visualforce markup, and find the page name that follows `src="$Page`.

   b. Remove the reference to the iframe from the markup. Be sure to make note of the Visualforce page in the iframe.

```
<apex:iframe src="{!$Page.AccountListView}"/>
```

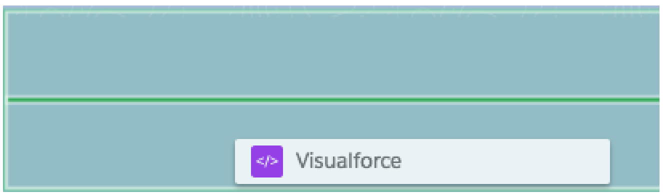
   c. Go into the properties of each Visualforce page by clicking `Edit`. Select the Available for Lightning Experience, Lightning Communities, and the mobile app checkbox. Check the checkbox for both Visualforce pages—the original page and the page embedded in the iframe.



Page Information

Label	AccountView
Name	AccountView
Description	
Available for Lightning Experience, Lightning Communities, and the mobile app	✓
Require CSRF protection on GET requests	☐

   d. Use the Lightning App Builder to drag two standard Visualforce components onto the canvas.



   e. Save and activate your Lightning Record Page.

These suggestions are meant to guide you in the right direction. If you're unable to find a working solution, post questions to the Lightning Exp Configuration Converter group in the Trailblazer Community. Use the #VisualforceInLightning hashtag to call on our experts.

Replace the window.location Method

The window.location variable is a JavaScript method used to redirect the browser to a new page. JavaScript code in a Visualforce page that sets window.location directly isn't compatible with Lightning Experience. That's the bad news. The good news is that sforce.one navigation methods do the same thing, and are compatible with Lightning Experience.

Because window.location can be used in many ways, it's not possible to address all of the possibilities. Let's look at a case where window.location is used as a link to the edit page for an account. Then you can use this example, including the code samples, as a guide for addressing other occurrences of window.location in your pages. If you can determine what your window.location method is trying to do, chances are a sforce.one workaround exists.

EDITIONS

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

USER PERMISSIONS

To create, edit, or delete Visualforce pages:

- Customize Application

1. Locate window.location in the Visualforce code.

   a. From Setup, enter *Visualforce* in the **Quick Find** box, then select **Visualforce Pages**.

   b. Click **Edit** next to the Visualforce page.

   c. Search the Visualforce markup for the following: *window.location*.

```
<apex:page standardController="Account"
lightningStylesheets="true">
  <apex:form >
    <apex:pageBlock >
      <apex:pageBlockSection title="Edit">
        <apex:commandButton value="Edit"
         onclick="window.location='/{!Account.Id}/e'; return
false;"/>
      </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
  <apex:detail />
</apex:page>
```

2. Determine where window.location is redirecting to. You can figure this out by testing the Visualforce page in Salesforce Classic.
3. Find the appropriate sforce.one method to replace window.location.
4. Replace all instances of window.location with sforce.one navigation.

```
<apex:page standardController="Account"
lightningStylesheets="true">
  <apex:form >
    <apex:pageBlock >
      <apex:pageBlockSection title="Edit">
        <apex:commandButton value="Edit"
         onclick="sforce.one.editRecord('{!Account.Id}');"/>
      </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
  <apex:detail />
</apex:page>
```

5. Save your changes.
6. If your Visualforce page is going to be accessed in both Salesforce Classic and Lightning Experience, make both options available. Use an if statement to branch the code.

```
<apex:page standardController="Account"
lightningStylesheets="true">
   <script>
     function isLightningExperience(){
        if (UITheme.getUITheme() === 'Theme4d' ||
UITheme.getUITheme() === 'Theme4u'){
            sforce.one.editRecord('{!Account.Id}');
        } else {
            window.location='/{!Account.Id}/e';
        }
    }
   </script>
   <apex:form >
     <apex:pageBlock >
       <apex:pageBlockSection title="Edit">
         <apex:commandButton value="Edit"
          onclick="javascript:isLightningExperience();return
false;"/>
       </apex:pageBlockSection>
     </apex:pageBlock>
   </apex:form>
   <apex:detail />
</apex:page>
```

These suggestions are meant to guide you in the right direction. If you're unable to find a working solution, post questions to the Lightning Exp Configuration Converter group in the Trailblazer Community. Use the #VisualforceInLightning hashtag to call on our experts.

Replace Static URLs

Lightning Experience doesn't support the use of static URLs to link to Salesforce resources. Fortunately, there's a much better alternative that future proofs your URLs. URLFOR is a function that calculates a URL based on inputs, rather than relying on something static that can't change. Use the URLFOR function with the appropriate action instead of direct URLs.

Because static URLs can be applied in many ways, it's not possible to cover all possibilities. Let's look at a case where a static URL is used to link to the edit page for an account. Then you can use this example, including the code samples, as a guide for addressing other static URLs in your pages.

EDITIONS

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

USER PERMISSIONS

To create, edit, or delete Visualforce pages:

- Customize Application

1. Locate href and .Id} in the Visualforce code.
   a. From Setup, enter `Visualforce` in the **Quick Find** box, then select **Visualforce Pages**.
   b. Click **Edit** next to the Visualforce page.
   c. Search the Visualforce markup for each of the following: `href` and `.Id}`.

```
<apex:page standardController="Account"
lightningStylesheets="true">
  <apex:outputLink value="/{!Account.Id}/e" >Edit
Account</apex:outputLink>
</apex:page>
```

2. Determine where the static URL is redirecting to. You can figure this out by testing the Visualforce page in Salesforce Classic.
3. Find the appropriate URLFOR action to replace window.location.

4. Replace all static URLs with URLFOR functions.

```
<apex:page standardController="Account"
lightningStylesheets="true">
  <apex:outputLink value="{!URLFOR($Action.Account.Edit,
Account.Id, null, true)}">Edit Account</apex:outputLink>
</apex:page>
```

5. Save your changes.

These suggestions are meant to guide you in the right direction. If you're unable to find a working solution, post questions to the Lightning Exp Configuration Converter group in the Trailblazer Community. Use the #VisualforceInLightning hashtag to call on our experts.

Apply Lightning Stylesheets to Visualforce Pages Manually

With Lightning stylesheets, it's easy to tweak your existing Visualforce pages so they'll display with classic styling in Salesforce Classic and Lightning styling in Lightning Experience.

EDITIONS

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

USER PERMISSIONS

To create, edit, or delete Visualforce pages:

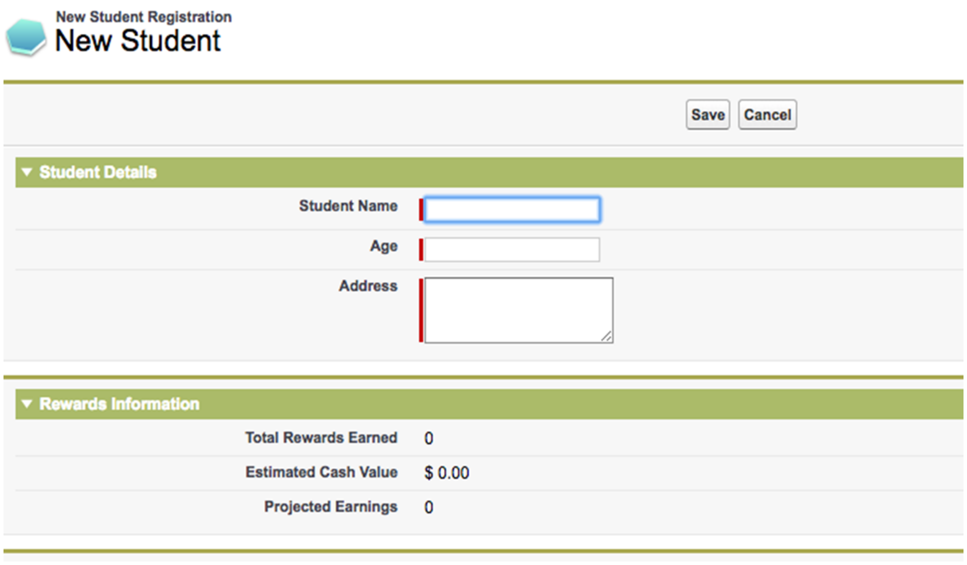
- Customize Application

1. From Setup, enter *Visualforce* in the **Quick Find** box, then select **Visualforce Pages**.
2. Click **Edit** next to the Visualforce page.
3. Add the *lightningStylesheets="true"* attribute to the initial *<apex:page>* component in the Visualforce markup.

```
<apex:page standardController="Account"
lightningStyleSheets="true">
```

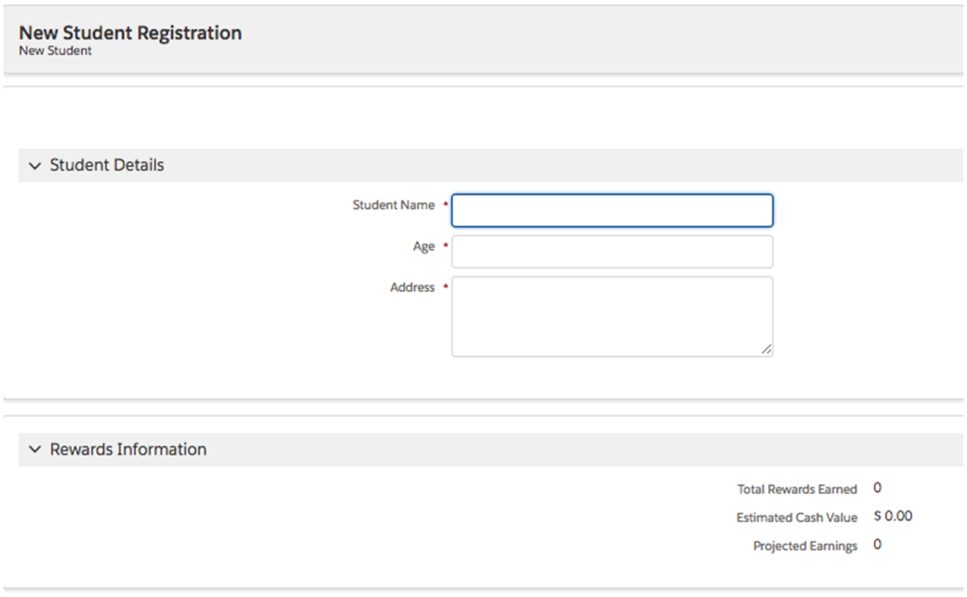
4. Save your changes.

When Lightning stylesheets are applied, here's how a Visualforce page looks in Salesforce Classic.



Student Name	
Age	
Address	
Total Rewards Earned	0
Estimated Cash Value	$ 0.00
Projected Earnings	0

And here's how the same page appears in Lightning Experience.



Student Name *	
Age *	
Address *	
Total Rewards Earned	0
Estimated Cash Value	$ 0.00
Projected Earnings	0

CHAPTER 15 Review AppExchange Packages for Lightning Experience Compatibility

Scan your installed package metadata and prepare your AppExchange packages for Lightning Experience. Find out which packages are ready to go. Get recommendations on updating, replacing, or verifying the rest.

Lightning Ready AppExchange packages work correctly within Lightning Experience.

Some AppExchange packages don't offer Lightning Ready versions. Some buttons, links, and Visualforce pages in these packages don’t work in Lightning Experience or aren’t visible to users. We sometimes suggest an alternative package that performs the same function and is Lightning Ready. If you have questions about additional configuration steps, functionality, Lightning Experience support, or recommended workarounds for a specific package, contact the package publisher.

EDITIONS

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

Available in: English language only

USER PERMISSIONS

To run the Lightning Experience Configuration Converter:
- Customize Application, Modify All Data

To convert JavaScript buttons and links or update hard-coded URLs:
- My Domain is deployed in your org

1. Log in to the Lightning Experience Configuration Converter, then select the **AppExchange** tab.
2. If your org hasn’t been scanned for AppExchange packages before, or if the scan data is out of date, scan to get the latest data.
3. Review the status of each package. Take action based on the status of each package.

- No update required: Your active package is ready for Lightning Experience. No further action is required.
- Update package: Your active package has a Lightning Experience-supported version available. Update the package to this version.
- Review suggested alternative: Your active package has a Lightning Experience-supported alternative available. The alternative isn't the same as your active package, but it performs the same function. Review and install this alternative package.
- No update or alternative: Your active package isn't ready for Lightning Experience and there's no suggested alternative. Contact the package publisher for a workaround.
- Unable to analyze: Your active package might be ready for Lightning Experience, but we weren't able to analyze it. Perform a manual check on the package to confirm that everything is ready for Lightning Experience.

4. Contact the package publisher in these cases.
   - To verify whether additional configuration steps are needed for the app to run in Lightning Experience.
   - If you're not sure whether the app is supported for Lightning Experience.
   - To learn about any recommended workarounds when the package isn't supported for Lightning Experience.

CHAPTER 16 Convert Attachments and Classic Notes

Does your org use attachments and notes? In Lightning Experience, users get a better experience if you convert existing attachments to Salesforce Files and classic notes to enhanced notes. You can automate this conversion work with the handy Magic Mover for Notes and Attachments tool.

EDITIONS

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

Note: This tool is a Salesforce Labs app. Salesforce Labs is a program that lets salesforce.com engineers, professional services staff, and other employees share AppExchange apps they've created with the customer community. Salesforce Labs apps are free to use, but are not official Salesforce products. They should be considered community projects, which aren't officially tested or documented. For help with a Salesforce Labs app, please consult the Salesforce message boards. Salesforce Support isn't available.

Install Magic Mover for Notes and Attachments from AppExchange.

CHAPTER 17 Migrate Your Classic Knowledge Base to Lightning Knowledge

If your company uses Classic Knowledge, Lightning Experience users must switch to Salesforce Classic to do things like find or create Knowledge articles. Moving your Classic knowledge base into Lightning Knowledge gives your users a seamless experience and much better workflow. When you're ready, the Lightning Knowledge Migration Tool does most of the heavy lifting for you.

CHAPTER 18 What Is a Lightning App?

In this chapter ...

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.

Similar to apps in Salesforce Classic, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. However, Lightning apps take things to a level beyond Classic apps. Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app.

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.

Let's jump into the details.

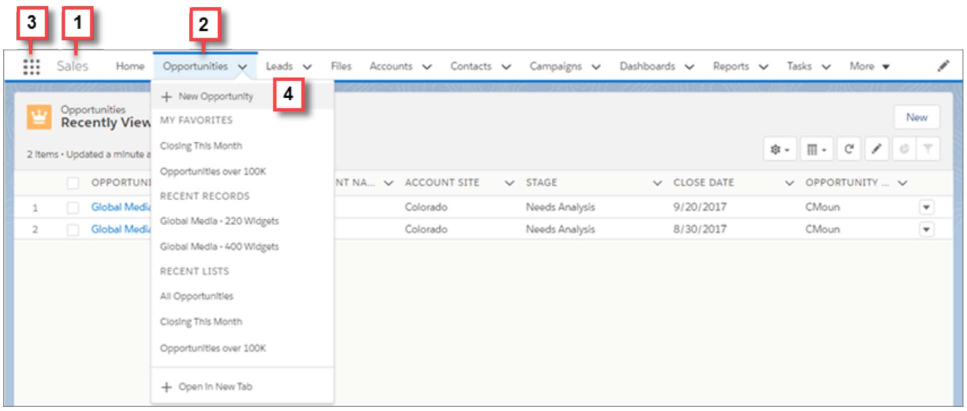
Navigate at the Speed of Lightning

If you know Salesforce Classic, the Lightning Experience navigation model will feel like a familiar friend, only better.

Each Lightning app has a navigation bar at the top of the page, letting your users:

- Find what they need using item names for easy recognition
- Complete actions and access recent records and lists with a single click
- Personalize the navigation bar to suit the unique way they work

Think of the navigation bar as a container for a set of items and functionality. It's always there, but the items within it change based on the app you're using.



	OPPORTUNI	NT NA...	ACCOUNT SITE	STAGE	CLOSE DATE	OPPORTUNITY ...
1	Global Medi		Colorado	Needs Analysis	9/20/2017	CMoun
2	Global Medi		Colorado	Needs Analysis	8/30/2017	CMoun

- The app name (1) displays on the left side of the navigation bar and custom colors and branding (2) make each app unique and easy to identify.
- Your users can access other items and apps by clicking the App Launcher icon (3).
- Your users can create records and access recent records and lists directly from the navigation bar (4) for items like Opportunities.

The items in an app are always just one click away in the navigation bar.

What can you put in the Lightning app navigation bar?

If you're familiar with Salesforce Classic, you know that Classic apps can contain:

- 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 can contain everything on this list plus Lightning page tabs and utilities like Lightning Voice. If your org uses utility features, you can enable a utility bar in your app that allows instant access to productivity tools, like integrated voice, in the Lightning Experience footer.

How can users personalize the navigation bar?

Personalized navigation in Lightning Experience is similar to customized tab sets in Salesforce Classic but better. In Lightning Experience, the navigation bar can contain more than just object-level items, like Accounts. You can add granular items, like a dashboard, list, or record. Let's look at the different ways users edit the navigation bar.

You can drag items on the navigation bar to change their order.

Note: The asterisk to the left of the item name indicates that it's a temporary tab. A temporary tab opens when you open an item that doesn't have a parent object already in the navigation bar. The temporary tab is removed from the nav bar when you close it, log out of Salesforce, or switch to a different app.

Want to make several changes to the navigation bar? Click the pencil icon on the navigation bar. From the enhanced editor:

- Reorder the items already in your navigation bar.
- Rename items you've added. A pencil icon is next to the items that you can rename
- Add items to the navigation bar. Click **Add More Items**. Search through your favorites or all available items in your org, and choose what to add. After you make your selections, you can reorder or remove items before saving your changes. You can't delete items that your admin has specified for the app.

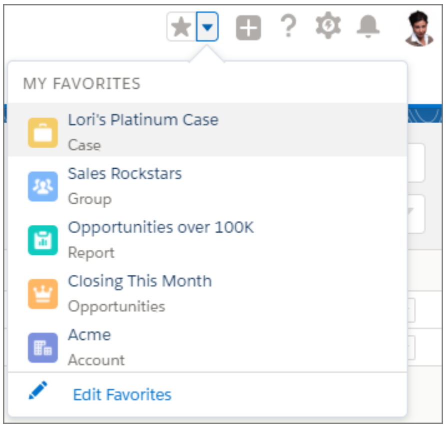
A few considerations to keep in mind.

- Help users get the most out of personalized navigation by upgrading your Classic apps to Lightning apps. Users can't personalize the navigation bar of Classic apps in Lightning Experience.
- Check what's in your apps. Users can't remove the items you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items.
- Items that you add to an app's navigation bar are added to the end of users' personalized navigation bars in the order that you added them.

- When you remove an item from an app, that item remains in your users' personalized navigation bars, and users can then delete it.
- If you don't want your users to personalize the navigation bar for a specific app, disable personalization. From Setup in Lightning Experience, go to the **App Manager**. For the desired app, select **App Options**. Select **Disable end user personalization of nav items in this app**.
- If you don't want your users to personalize the navigation bar for any app, disable personalization. From Setup, enter `User Interface` in the Quick Find box, then select **User Interface**. Select **Disable Navigation Bar Personalization in Lightning Experience**. Salesforce recommends disabling navigation personalization by app instead of for the entire org.
- Control if temporary tabs are created when users access items outside of the app. From Setup in Lightning Experience, go to the **App Manager**. Edit the desired app. On the **App Options** page, select **Disable temporary tabs for items outside of this app**.
- The tab's dropdown menu includes an option to create an item. For example, create an account by selecting **+ New Account** from the Account tab's dropdown menu. However, the New action doesn't appear if:
  - Users don't have the create permission for the object
  - The New action isn't part of the search layout for the object's list view

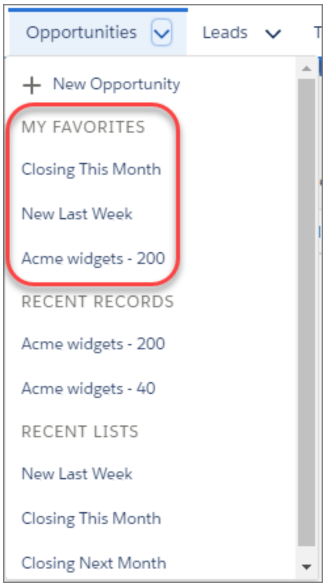
Fast Track Navigation in Lightning Experience with Favorites

Favorites are another great way for your users to boost productivity by personalizing how they navigate. Think of the favorites star at the top of the page in Lightning Experience as a constellation in the sky. It's always there, so your users can easily navigate to what they need no matter where they are.



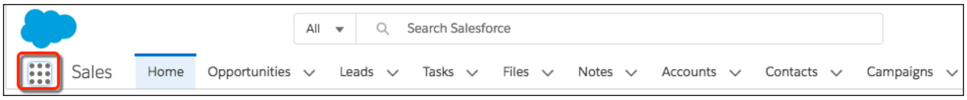
Click the favorites star to add the current page—for example, opportunities closing this month—to your favorites. Saved favorites are always just a click away in the favorites list. A highlighted star means you're on a favorite page. To remove a favorite, click the highlighted star. Your users can easily personalize their favorites lists by renaming favorites and rearranging items in the list.

If an item in the navigation bar has a dropdown menu, it automatically includes the user's top favorites for the item.

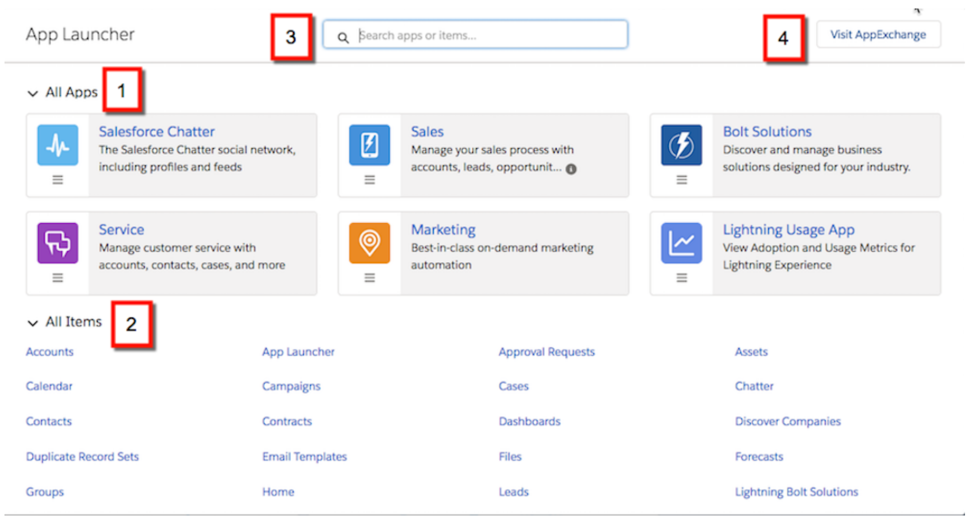


Find Items and Apps in the App Launcher

With the App Launcher, your users can switch between apps and access available items and features. The App Launcher is so central to navigating Lightning Experience and its apps, we've positioned it within easy reach on the left side of the navigation bar. It's available from any page.



The App Launcher displays all available apps in one place so your users can easily find what they're looking for—even apps they didn't know they had.

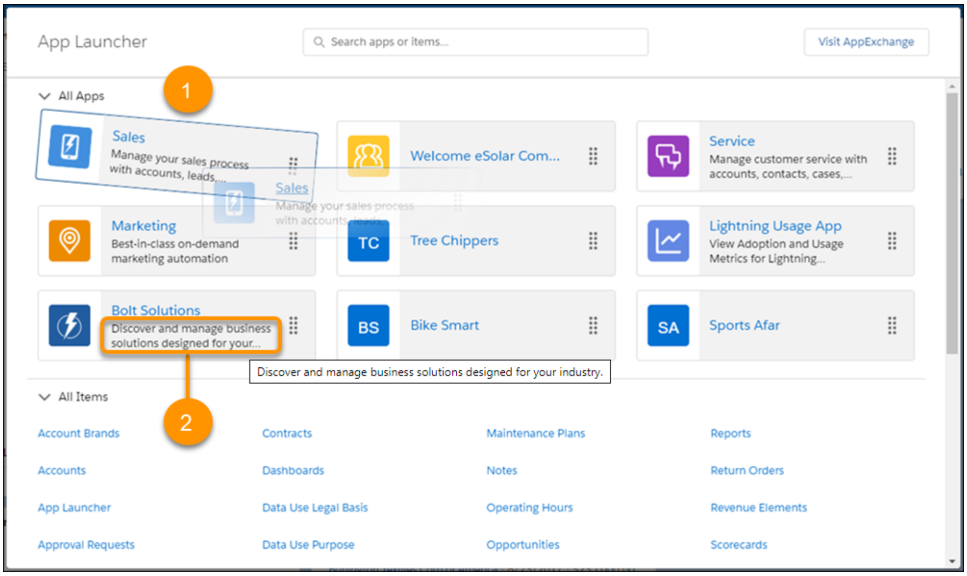

App Launcher
3
Search apps or items...
4
Visit AppExchange
All Apps
1
Salesforce Chatter
The Salesforce Chatter social network, including profiles and feeds
Sales
Manage your sales process with accounts, leads, opportunit...
Bolt Solutions
Discover and manage business solutions designed for your industry.
Service
Manage customer service with accounts, contacts, cases, and more
Marketing
Best-in-class on-demand marketing automation
Lightning Usage App
View Adoption and Usage Metrics for Lightning Experience
All Items
2
Accounts
App Launcher
Approval Requests
Assets
Calendar
Campaigns
Cases
Chatter
Contacts
Contracts
Dashboards
Discover Companies
Duplicate Record Sets
Email Templates
Files
Forecasts
Groups
Home
Leads
Lightning Bolt Solutions

Apps show up in the App Launcher as large tiles under All Apps (1). Other items, such as custom objects, tasks, events, and the feed, show up under All Items (2). Don't see what you want? Search for it by name in the search box (3). Expand your search to find the latest cloud-computing apps and services available on the AppExchange (4). You can do it all with the click of a button without ever leaving Lightning Experience.

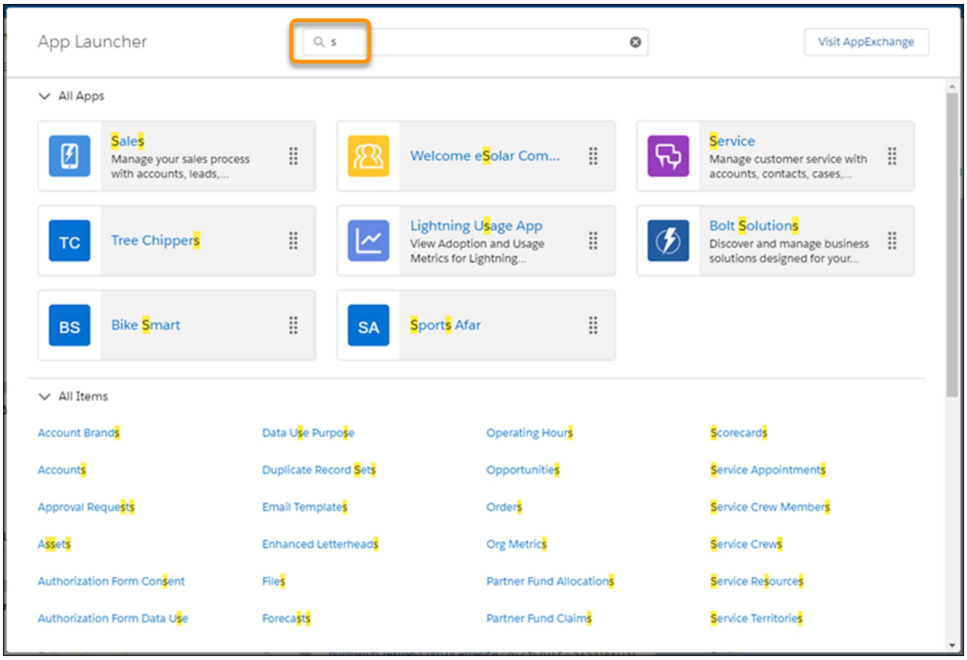
You can make various types of apps available to your users in the App Launcher, including:

- Salesforce apps, which include custom apps and standard apps that come with Salesforce, like Sales and Service
- Connected apps such as Gmail™ and Microsoft® Office 365™
- Partner and ISV apps

As an admin, you can change which apps appear on the App Launcher and the default order in which they appear. You and your users can then drag the tiles around to create your own personal view of the App Launcher. Your users see only the apps you authorize for them to access through profiles or permission sets.



The App Launcher's great for finding an app or item even when it's not currently on the navigation bar. Just open the App Launcherand search for it by name. For example, say you're looking for an item called Service. Enter `Service` in the search box to see items and apps that match your search as you type.



App Launcher

s

Visit AppExchange

All Apps

Sales
Manage your sales process with accounts, leads,...

Welcome eSolar Com...

Service
Manage customer service with accounts, contacts, cases,...

Tree Chippers

Lightning Usage App
View Adoption and Usage Metrics for Lightning...

Bolt Solutions
Discover and manage business solutions designed for your...

Bike Smart

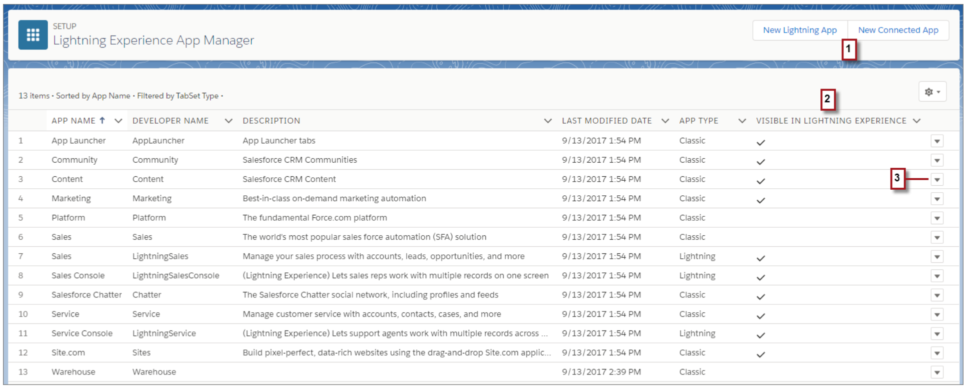
Sports Afar

All Items

Account Brands	Data Use Purpose	Operating Hours	Scorecards
Accounts	Duplicate Record Sets	Opportunities	Service Appointments
Approval Requests	Email Templates	Orders	Service Crew Members
Assets	Enhanced Letterheads	Org Metrics	Service Crews
Authorization Form Consent	Files	Partner Fund Allocations	Service Resources
Authorization Form Data Use	Forecasts	Partner Fund Claims	Service Territories

Meet the Lightning Experience App Manager

The App Manager in Setup is your go-to place for managing apps for Lightning Experience. It shows all your connected apps and Salesforce apps, both Classic and Lightning.



SETUP
Lightning Experience App Manager

New Lightning App | New Connected App

1

13 items • Sorted by App Name • Filtered by TabSet Type •

2

3

	APP NAME	DEVELOPER NAME	DESCRIPTION	LAST MODIFIED DATE	APP TYPE	VISIBLE IN LIGHTNING EXPERIENCE
1	App Launcher	AppLauncher	App Launcher tabs	9/13/2017 1:54 PM	Classic	✓
2	Community	Community	Salesforce CRM Communities	9/13/2017 1:54 PM	Classic	✓
3	Content	Content	Salesforce CRM Content	9/13/2017 1:54 PM	Classic	✓
4	Marketing	Marketing	Best-in-class on-demand marketing automation	9/13/2017 1:54 PM	Classic	✓
5	Platform	Platform	The fundamental Force.com platform	9/13/2017 1:54 PM	Classic	
6	Sales	Sales	The world's most popular sales force automation (SFA) solution	9/13/2017 1:54 PM	Classic	
7	Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	9/13/2017 1:54 PM	Lightning	✓
8	Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one screen	9/13/2017 1:54 PM	Lightning	✓
9	Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	9/13/2017 1:54 PM	Classic	✓
10	Service	Service	Manage customer service with accounts, contacts, cases, and more	9/13/2017 1:54 PM	Classic	✓
11	Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records across ...	9/13/2017 1:54 PM	Lightning	✓
12	Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com applic...	9/13/2017 1:54 PM	Classic	✓
13	Warehouse	Warehouse		9/13/2017 2:39 PM	Classic	

You can use the Lightning Experience App Manager to:

- Create Lightning apps or connected apps (1)
- See if your Classic apps are accessible to your users in Lightning Experience (2)
- Edit, delete, or upgrade Classic apps to take advantage of all the benefits of apps in Lightning Experience (3)

Tip: Click a column header to sort the list based on that column.

What does that "Visible in Lightning Experience" column mean?

A checkmark in the Visible in Lightning Experience column means that the app is accessible in Lightning Experience via the App Launcher and is fully functional.

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. Classic apps that have been upgraded to Lightning apps automatically have that setting disabled.

We've mentioned a couple of times now that you can upgrade your Classic apps. Why do that? So your apps can take advantage of all the perks of Lightning apps. We'll get there shortly. But first, let's get your org ready for it.

Install a Custom Classic App: The Warehouse Data Model

We want to walk you through upgrading a Classic app to Lightning. To do that, you need to have a clean custom Classic app in your org. So we're going to enlist an old friend for help: the enhanced Warehouse data model. You might be familiar with the Warehouse app if you've gone through the tutorials in the *Lightning Platform Workbook*. We took the Warehouse data from that guide and added a few extra things to it. If you've gone through the exercises in the *Salesforce Mobile App Developer Guide*, you probably already have the enhanced Warehouse app installed.

Important: Check your Developer org. If you already have the Warehouse app installed, skip these steps.

1. Go to `www.salesforce.com` and log in to Salesforce using your Developer Edition credentials.
2. Open a new browser tab or window, and navigate to `https://github.com/forcedotcom/Salesforce1-Dev-Guide-Setup-Package`.

Do this in the browser that you logged in to your Developer org with.

3. Open the README file.
4. Click the link in the README file.

   This is the installation link for the enhanced Warehouse data package. You should be taken directly to the Package Installation Details page.
5. Click **Install**.
6. Wait for the installation to finish, then click **Done**.
7. If your DE org is set to Lightning Experience, switch to Salesforce Classic.
8. From the app menu, select **Warehouse**.
9. Click the **Data** tab.
10. Click **Create Data**.
11. Click your name in the upper-right corner, then select **Switch to Lightning Experience**.

CHAPTER 19 Step into the World of Lightning Apps

In this chapter ...

Creating and editing a Lightning app is a cinch. 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 important items (including tabs) that 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.

Tip: If your org includes utilities like Lightning Voice or Open CTI Softphone, you can add them to your app's utility bar.

Create a Lightning App

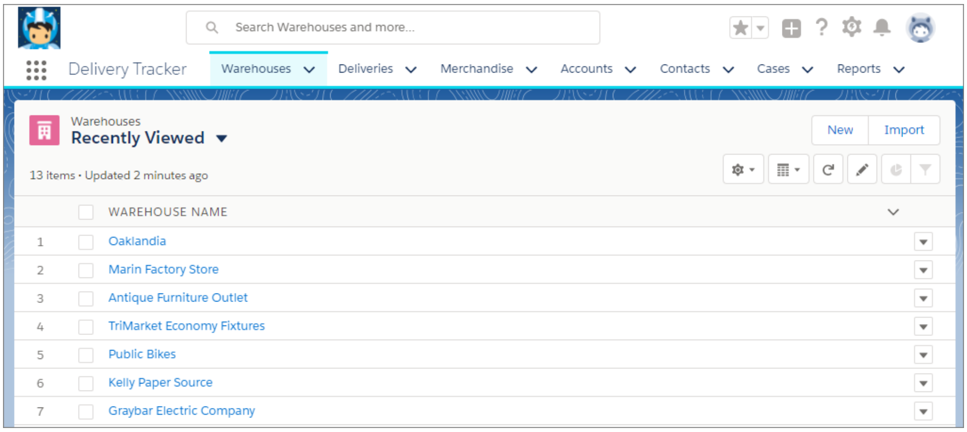
You've got the Warehouse app installed in your org. Let's create a mini app for the warehouse delivery managers, who want to know what inventory is getting delivered where, whether it got there on time, and what cases related to deliveries are open.

1. From the Home tab in Setup, enter `App` in the `Quick Find` box, then select **App Manager**.
2. Click **New Lightning App**.
3. In the Lightning App Wizard, create an app with these parameters.

App name	Delivery Tracker
Description	Track warehouse deliveries.
Image	Your choice! Use 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.
Primary hex color value	#09D4EA
App navigation	Standard
Navigation bar items (in this order)	Warehouses, Deliveries, Merchandise, Accounts, Contacts, Cases, Reports
Assigned to user profile	System Administrator

4. Click **Save and Finish** to exit the wizard.
5. From the App Launcher (⠿), find and select **Delivery Tracker**.
6. Check out the new app!

   It's got all the custom branding we gave it: a custom icon in the upper left and the custom color we assigned to it. Because Warehouses is first in the navigation bar, it becomes the landing page for the app. And even though App Launcher was an item available for the navigation bar, we didn't have to add it. It's there, accessible by clicking ⠿ .



13 items • Updated 2 minutes ago

	WAREHOUSE NAME
1	Oaklandia
2	Marin Factory Store
3	Antique Furniture Outlet
4	TriMarket Economy Fixtures
5	Public Bikes
6	Kelly Paper Source
7	Graybar Electric Company

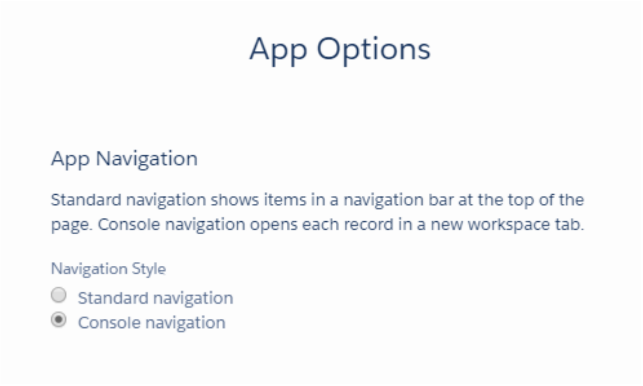
Nice work! Now you're ready to create your own custom Lightning apps.

Beyond the Basics

Did you know that app images for Lightning apps can be animated GIFs? Oh yes, they can. You're welcome.

Create Apps for Different Audiences

When you create a Lightning app, you get to decide which navigation style is best for your users and what important items belong in the app.


App Options

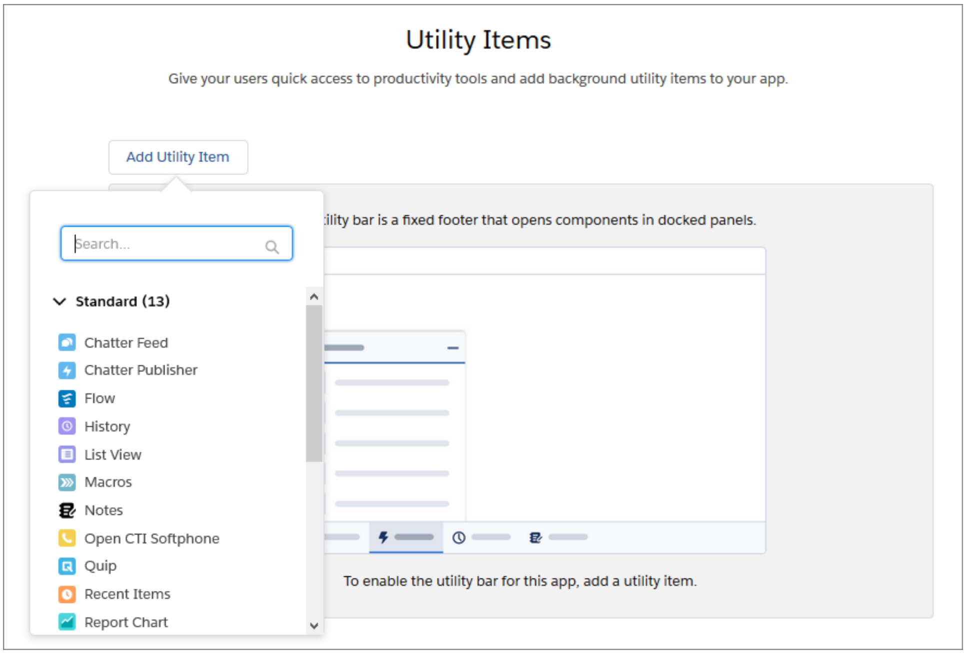
App Navigation

Standard navigation shows items in a navigation bar at the top of the page. Console navigation opens each record in a new workspace tab.

Navigation Style

Standard navigation [✓ Console navigation]

So which navigation style is best—standard or console? As with most things, it depends. Service users often work best using console navigation because it shows more information in a single view. Console navigation also provides quick access to standard and custom Lightning components, like integrated telephony, in an optional utility bar. And while console navigation is geared toward service users, other teams like marketing, development, and sales teams can also find it helpful. For example, sales teams can work from phone lead queues or get incoming leads through Omni-Channel, which are utility bar features available only in console navigation.


Utility Items

Give your users quick access to productivity tools and add background utility items to your app.

ility bar is a fixed footer that opens components in docked panels.

To enable the utility bar for this app, add a utility item.

After deciding on the navigation style, your next step is to decide what belongs in the app.

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.

CHAPTER 20 Take Your Classic App to the Next Level: Lightning

In this chapter ...

Classic apps work in Lightning Experience. So why upgrade them to Lightning apps? Upgrading a Classic app to a Lightning app lets you and your users take advantage of custom branding and the enhanced navigation features available in Lightning Experience.

Note: You can't upgrade a Salesforce Classic console app to Lightning Experience. You can choose to display or hide the app in the Lightning Experience App Launcher, but you can't edit the app from the App Manager page in Lightning Experience Setup. To get started in Lightning Experience, customize these Salesforce-provided Lightning console apps: Service Console and Sales Console. You can also recreate your Salesforce Classic console app in Lightning Experience, but using Salesforce's out-of-the-box app is faster and easier.

You installed the enhanced Warehouse app into your org, but it installed as a Classic app. Let's bring it into Lightning.

Upgrade the Warehouse App

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. Let's do that now.

1. From the Home tab in Setup, enter `App` in the `Quick Find` box, then select **App Manager**.
2. Find the Warehouse app in the list.
3. Click ▾ from the Warehouse app row, and select **Upgrade**.
4. Leave the suggested "Warehouse Lightning" name as-is, and click **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 still see the Classic app in the apps list, but with the Visible in Lightning column deselected.

After you upgrade a Classic app, the two versions of the app 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.

Give the Warehouse App That Touch of Lightning

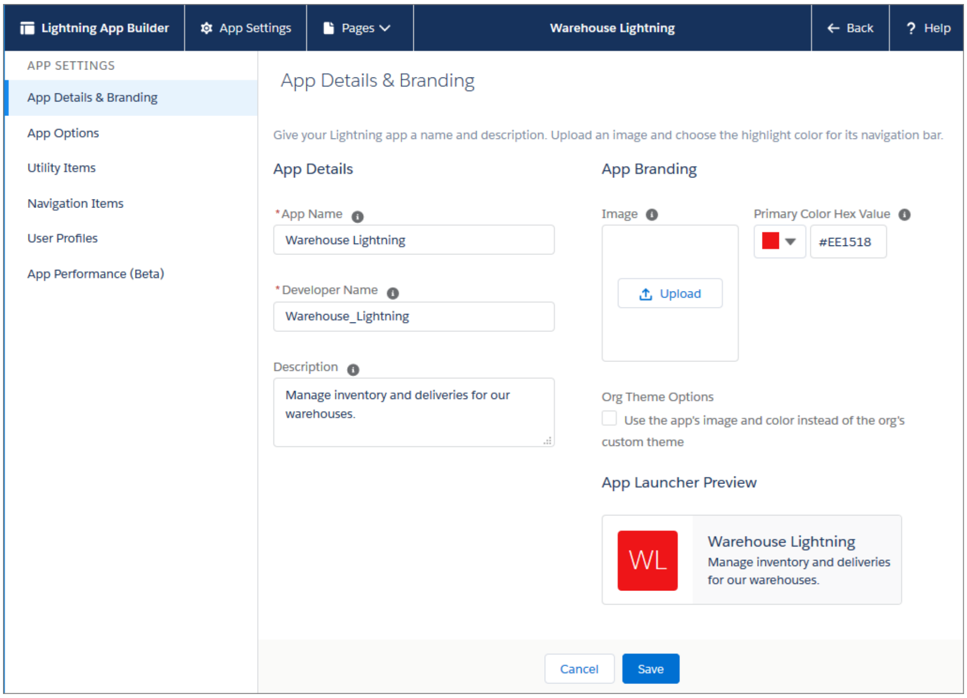
You've upgraded the Warehouse app, but it's not taking advantage of the great enhancements that Lightning apps can offer. Let's fix that.

1. Find the Warehouse Lightning app in the list.
2. Click ▾ from the Warehouse Lightning app row, and select **Edit**.

   The app opens inside the Lightning App Builder. Not only can you see all your app options, but you can view any active Lightning pages assigned to the app.
3. Update the description to *`Manage inventory and deliveries for our warehouses.`*

   The app description displays alongside the icon in the App Launcher, so it's a great way to tell your users what the app is for, but keep it brief.
4. In the App Branding section, change the Primary Color Hex Value to *`#EE1518`*.

   You can see a preview of the default app icon using the red color we assigned and what the app tile will look like in the App Launcher.



App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Warehouse Lightning

*Developer Name: Warehouse_Lightning

Description: Manage inventory and deliveries for our warehouses.

App Branding

Image: Upload

Primary Color Hex Value: #EE1518

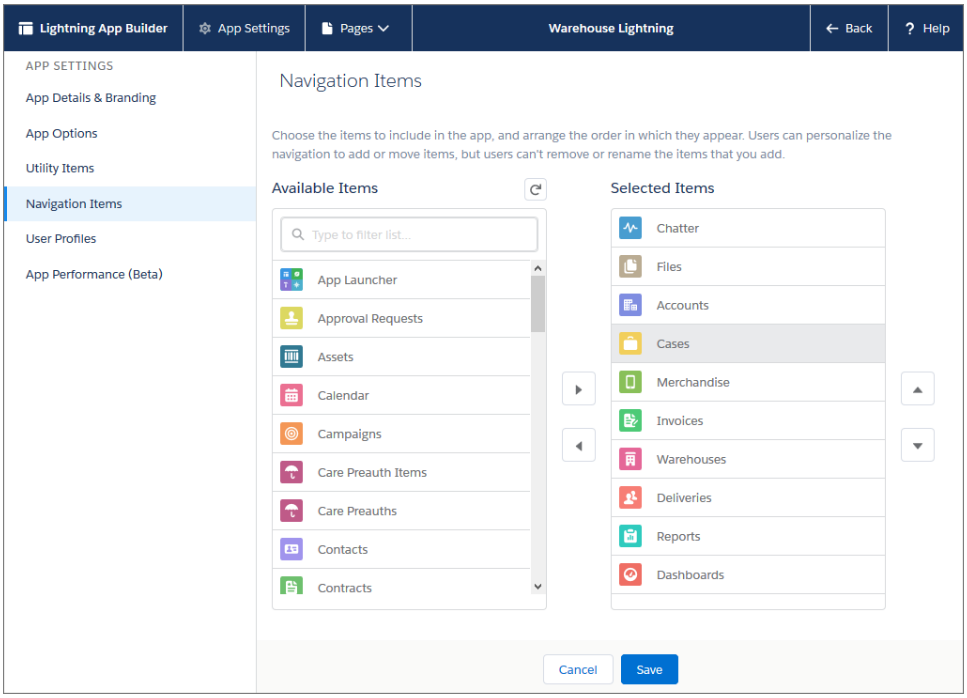
Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview

WL Warehouse Lightning
Manage inventory and deliveries for our warehouses.

5. Click **Save**.
6. On the Navigation Items tab, remove the Data and Home tabs from the Selected Items list. That leaves the Chatter tab at the top of the list, which means it becomes the app's landing page.
7. Add Cases to the Selected Items list, and move it to directly after Accounts.



Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add.

Available Items	Selected Items
App Launcher	Chatter
Approval Requests	Files
Assets	Accounts
Calendar	Cases
Campaigns	Merchandise
Care Preauth Items	Invoices
Care Preauths	Warehouses
Contacts	Deliveries
Contracts	Reports
	Dashboards

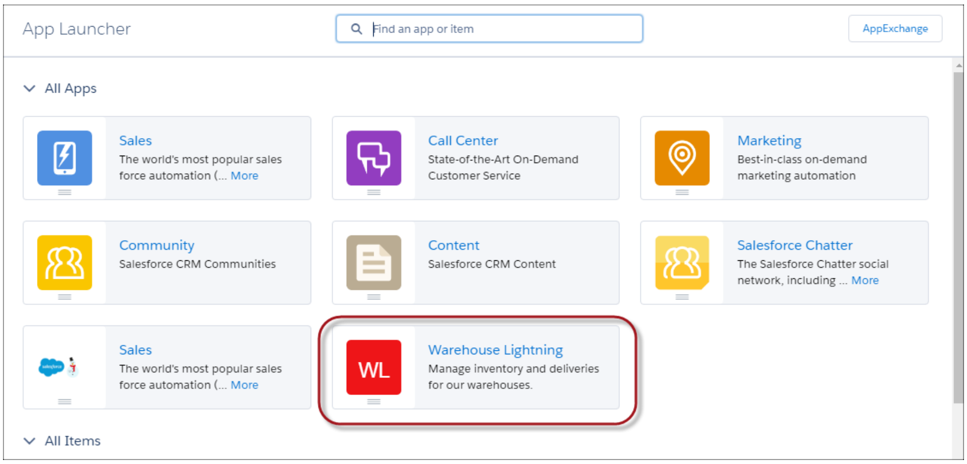
Tip: If you have Lightning pages in your org, you can see them in the Available Items list. Being able to add Lightning pages to the navigation bar is another great perk about Lightning apps.

8. Click **Save**.
9. On the User Profiles tab, make sure that System Administrator is in the Selected Items list.
10. Click **Save**, and then click **Back** to return to Setup.

Look at the Warehouse App Now

Let's look at the changes we made to the Warehouse app.

1. Open the App Launcher (⠿), and click **View All**.
   There's the Warehouse Lightning app.


App Launcher

Find an app or item

AppExchange

All Apps

Sales
The world's most popular sales force automation (... More

Call Center
State-of-the-Art On-Demand Customer Service

Marketing
Best-in-class on-demand marketing automation

Community
Salesforce CRM Communities

Content
Salesforce CRM Content

Salesforce Chatter
The Salesforce Chatter social network, including ... More

Sales
The world's most popular sales force automation (... More

Warehouse Lightning
Manage inventory and deliveries for our warehouses.

All Items

2. Click the Warehouse Lightning app.
   If the "Welcome to Salesforce" Setup Assistant dialog appears, close it.

   You're in the updated Warehouse app. It's got Chatter as the landing page, Cases right after Accounts, and the bright red color we gave it.

Good job! You've got the skills. Now go create and update some apps in your own org.

CREATE CUSTOM PAGES FOR LIGHTNING EXPERIENCE

CHAPTER 21 Meet the Lightning App Builder

In this chapter ...

Your users are busy. They're closing deals, providing top-notch service, and marketing to your prospects and customers. By creating customized pages, you can put key information at your users' fingertips and provide them with an easy interface to update and add records.

The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place.

But that's not all. When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. You can update the app's branding, navigation, options, and manage the Lightning pages assigned to that app all in the Lightning App Builder.

In this module, we'll be exploring Lightning page creation.

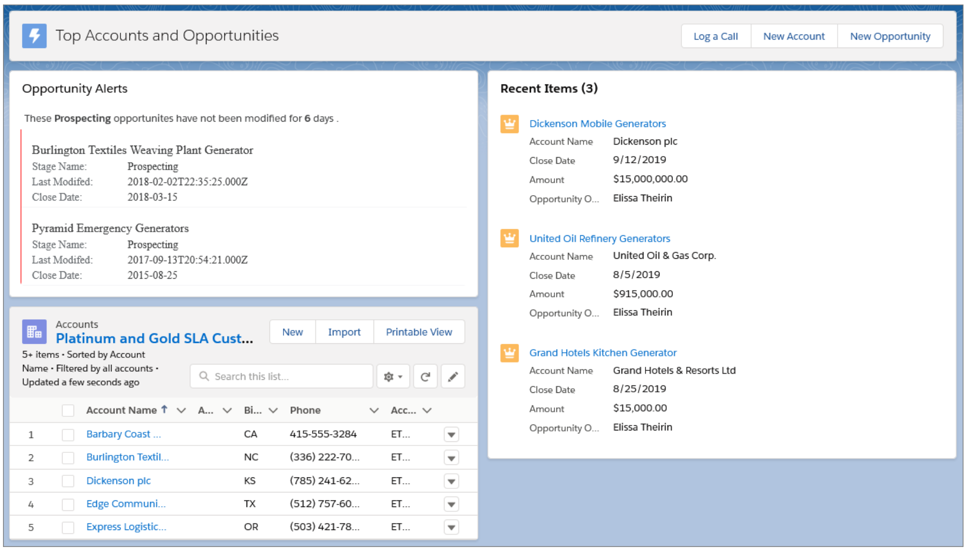
How the Lightning App Builder Works

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

A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or Lightning Experience. A Lightning page is composed of regions that contain components.

Here's a sneak peek at one of the pages you're going to build.



Opportunity Alerts

These **Prospecting** opportunites have not been modified for **6** days .

Burlington Textiles Weaving Plant Generator
Stage Name: Prospecting
Last Modifed: 2018-02-02T22:35:25.000Z
Close Date: 2018-03-15

Pyramid Emergency Generators
Stage Name: Prospecting
Last Modifed: 2017-09-13T20:54:21.000Z
Close Date: 2015-08-25

Accounts
Platinum and Gold SLA Cust...
5+ items • Sorted by Account Name • Filtered by all accounts • Updated a few seconds ago

	Account Name ↑	A...	Bi...	Phone	Acc...
1	Barbary Coast ...		CA	415-555-3284	ET...
2	Burlington Textil...		NC	(336) 222-70...	ET...
3	Dickenson plc		KS	(785) 241-62...	ET...
4	Edge Communi...		TX	(512) 757-60...	ET...
5	Express Logistic...		OR	(503) 421-78...	ET...

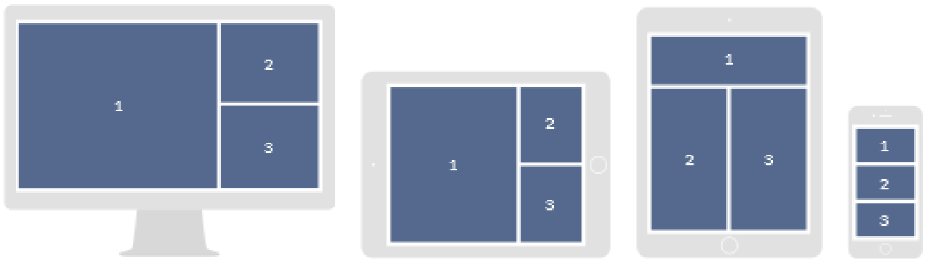
Recent Items (3)

Dickenson Mobile Generators
Account Name Dickenson plc
Close Date 9/12/2019
Amount $15,000,000.00
Opportunity O... Elissa Theirin

United Oil Refinery Generators
Account Name United Oil & Gas Corp.
Close Date 8/5/2019
Amount $915,000.00
Opportunity O... Elissa Theirin

Grand Hotels Kitchen Generator
Account Name Grand Hotels & Resorts Ltd
Close Date 8/25/2019
Amount $15,000.00
Opportunity O... Elissa Theirin

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 Components

A Lightning component is a compact, configurable, and reusable element that you can add to a Lightning page in the Lightning App Builder.

Lightning pages support these components:

Standard Components

Standard components are Lightning components built by Salesforce.

Custom Components

Custom components are Lightning components that you or someone else have created. You can configure custom Lightning components to work in Lightning App Builder.

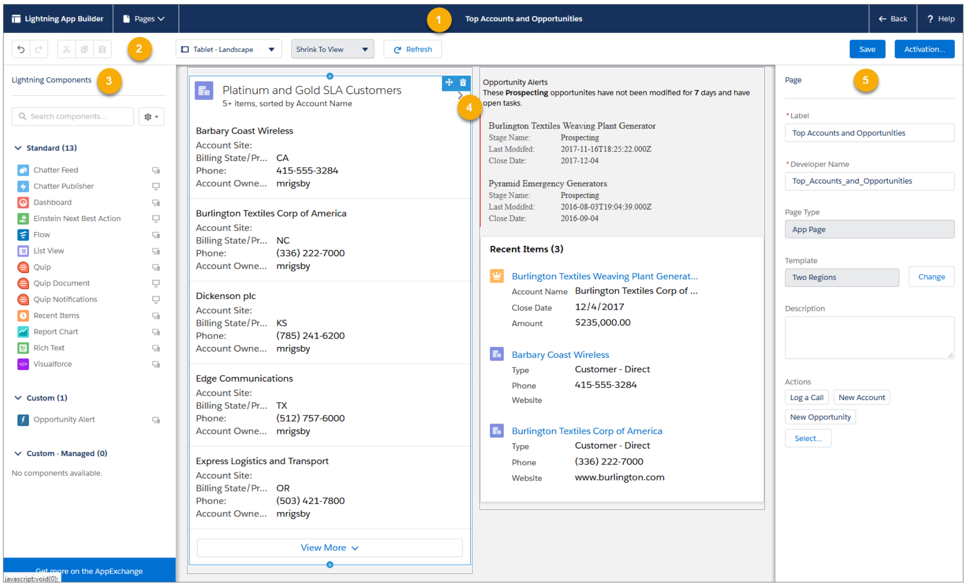
Third-Party Components on AppExchange

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.

Note: As of the Spring '19 release (API version 45.0), you can build Lightning components using two programming models: the Lightning Web Components model and the original Aura Components model. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page.

The Lightning App Builder User Interface

The Lightning App Builder's user interface makes creating Lightning pages easy. Here's a breakdown of the parts of the tool.



Platinum and Gold SLA Customers
5+ items, sorted by Account Name

Barbary Coast Wireless	
Account Site:	
Billing State/Pr...	CA
Phone:	415-555-3284
Account Owne...	mrigsby
Burlington Textiles Corp of America	
Account Site:	
Billing State/Pr...	NC
Phone:	(336) 222-7000
Account Owne...	mrigsby
Dickenson plc	
Account Site:	
Billing State/Pr...	KS
Phone:	(785) 241-6200
Account Owne...	mrigsby
Edge Communications	
Account Site:	
Billing State/Pr...	TX
Phone:	(512) 757-6000
Account Owne...	mrigsby
Express Logistics and Transport	
Account Site:	
Billing State/Pr...	OR
Phone:	(503) 421-7800
Account Owne...	mrigsby

Opportunity Alerts
These **Prospecting** opportunites have not been modified for **7** days and have open tasks.

Burlington Textiles Weaving Plant Generator
Stage Name: Prospecting
Last Modifed: 2017-11-16T18:25:22.000Z
Close Date: 2017-12-04

Pyramid Emergency Generators
Stage Name: Prospecting
Last Modifed: 2016-08-03T19:04:39.000Z
Close Date: 2016-09-04

Recent Items (3)

Burlington Textiles Weaving Plant Generat...
Account Name Burlington Textiles Corp of ...
Close Date 12/4/2017
Amount $235,000.00

Barbary Coast Wireless
Type Customer - Direct
Phone 415-555-3284
Website

Burlington Textiles Corp of America
Type Customer - Direct
Phone (336) 222-7000
Website www.burlington.com

*Label: Top Accounts and Opportunities
*Developer Name: Top_Accounts_and_Opportunities
Page Type: App Page
Template: Two Regions — Change
Description:

Header (1)

When you're working on a Lightning page, the header shows you its label, and contains a Pages list where you can see the last 10 pages that you modified. You can also return to Setup without saving or to view more help for the Lightning App Builder.

If you're editing an app, the header also shows the app name and contains an App Settings tab where you can configure the app's options such as branding, navigation, and the utility bar. In an app context, the Pages list shows all active Lightning pages associated with the current app.

Toolbar (2)

Use the buttons in the toolbar to cut ( ), copy ( ), and paste ( ) page content; and to undo ( ), redo ( ), save, or activate your Lightning page. You can also view your page in different formats, refresh the canvas, or adjust the canvas size to fit your view.

Lightning Components Pane (3)

The components pane contains all standard and custom Lightning components that are supported for your Lightning page. Click and drag a component to add it to the page.

Tip: If you have a lot of custom components, enter text in the search field to easily find the one you need. You can access third-party custom components on the AppExchange using the button at the bottom of the pane.

Lightning Page Canvas (4)

The canvas area is where you build your page. Drag components to reorder them on the page.

Properties Pane (5)

Depending on what you select on the page, the properties pane shows either the overall page properties or the properties of the component that you've selected. Click **Page** in the breadcrumb to access the page properties when viewing a component.

Lightning Page Types

You can create different types of Lightning pages with the Lightning App Builder. We're going to look at these three types.

App Page

Use an app page to create a home page for a third-party app that you can add directly into the Salesforce mobile app and Lightning Experience navigation menus. Your users then have an app home page where they can quickly access the most important objects and items.

Home Page

Create Home pages with features relevant to specific types of users, and assign the customized pages to different apps or app-and-user-profile combinations. Custom Home pages are supported in Lightning Experience only.

Record Page

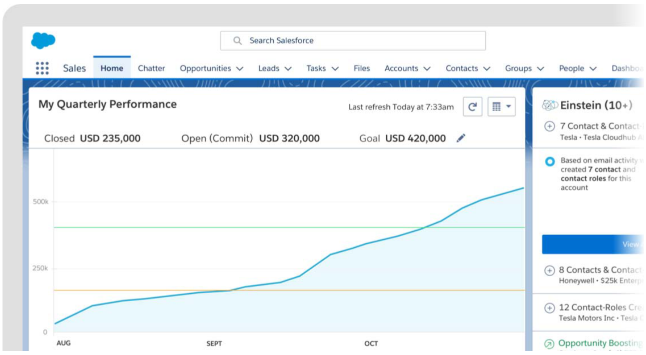
With a record page, you can create a customized version of an object's record page, tailoring it to your users' needs. Custom record pages are supported in Lightning Experience and in the Salesforce mobile app.

Let's start with the Home page.

CHAPTER 22 Yes, Virginia, You Can Customize Your Lightning Experience Home Page

In this chapter ...

Give your users everything they need to manage their day from the Home page in Lightning Experience. Your sales reps can see their quarterly performance summary and get important updates on critical tasks and opportunities. You can also customize the page for different types of users and assign custom pages to different apps and app-and-profile combinations.



You can create a custom Home page in a few different ways: Create it from scratch using a template, clone it from another custom Home page, or edit a page from the Home tab in a Lightning app.

To edit an existing page, you can click from the Home page, and then select **Edit Page** to create a copy of the current page to edit. If a customized page exists and is active, selecting Edit Page opens that page to edit.

We're going to create a Home page from scratch.

Beyond the Basics

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. Pages created like this—as copies—retain a reference to the standard page, which means that Salesforce can upgrade the page copies for you with new capabilities in the future. If you create a page from scratch, then you own it completely and new page capabilities Salesforce rolls out don't appear automatically.

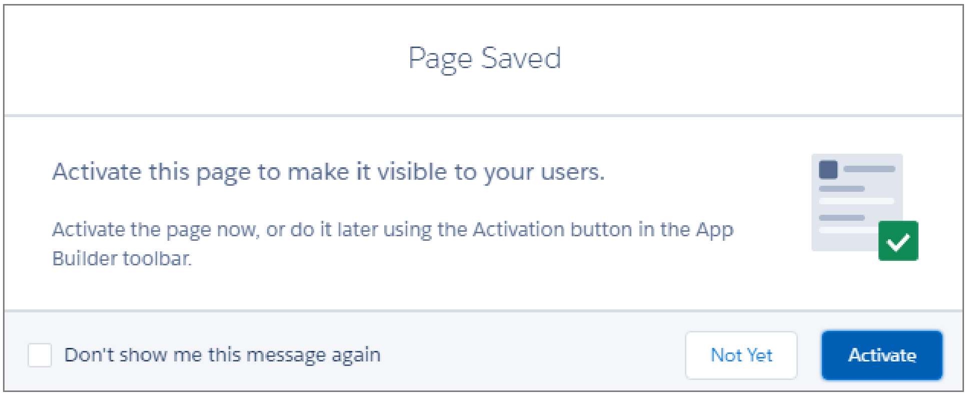
Create a Custom Home Page for Lightning Experience

We'll tweak the position of the components on the standard Home page layout slightly to give you an idea of what's possible.

Let's get started!

1. From Setup, enter `App Builder` in the Quick Find box, then click **Lightning App Builder**.
2. Click **New**, select **Home Page**, then click **Next**.
3. Step through the wizard and name the page `New Home Page`, select the Standard Home Page template, and then click **Finish**.

   An empty Home page opens, ready for you to build it. The components pane contains all the standard components available for a Home page, plus any custom components you have installed in your org.
4. Drag the Assistant component to the top-right region.
5. Drag Performance to the top-left region.
6. Add Today's Events to the lower left region and add Key Deals to the lower right region.
7. Add Today's Tasks above the Assistant.
8. Click **Save**.


Page Saved

Activate this page to make it visible to your users.

Activate the page now, or do it later using the Activation button in the App Builder toolbar.

But wait, what's this? There's more? Yes, there is. Saving the page isn't enough to get it out to your users. Lightning pages must be activated before your users can see them.

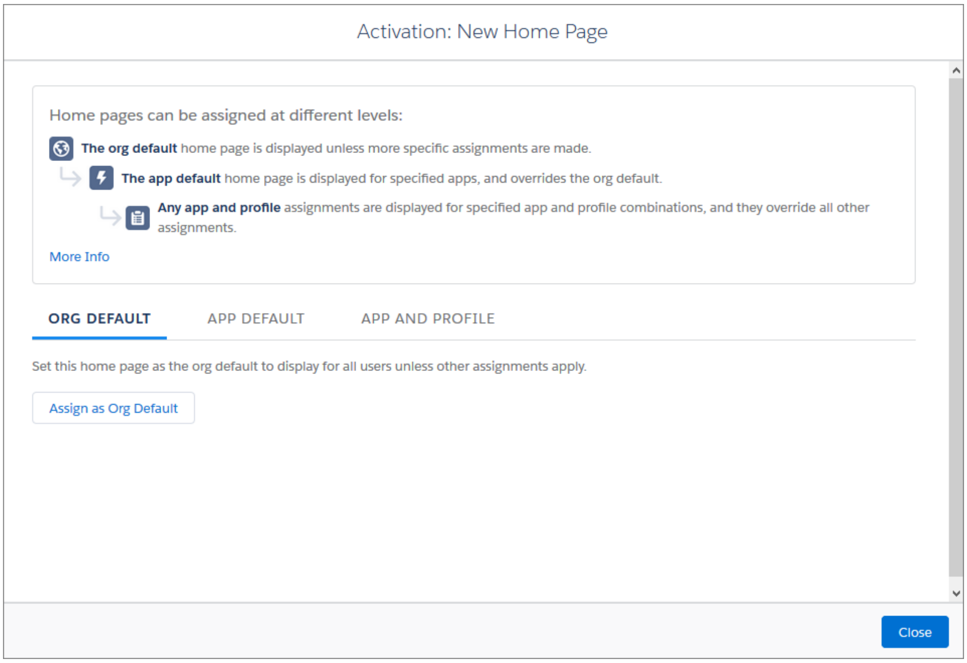
Normally, if you aren't done with your page, or aren't ready to make it public, you can click **Not Yet** here to save the page and return to the App Builder. But that's not us. We're bold! We're done with our page and want to give it to our users right now!

9. Click **Activate**, and we'll do just that.

   Tip: If you saved previously and didn't activate the page, you can click the **Activation** button in the toolbar to be ready for the next section.

Roll Out Your Custom Home Page to Your Lightning Experience Users

When activating a Home page, you have three different options: You can make your page the default for everyone in the org, the default for an app, or assign it to one or more app and profile combinations, giving your users access to a page designed just for their role.



Activation: New Home Page

Home pages can be assigned at different levels:

The org default home page is displayed unless more specific assignments are made.

The app default home page is displayed for specified apps, and overrides the org default.

Any app and profile assignments are displayed for specified app and profile combinations, and they override all other assignments.

More Info

ORG DEFAULT | APP DEFAULT | APP AND PROFILE

Set this home page as the org default to display for all users unless other assignments apply.

Assign as Org Default

Close

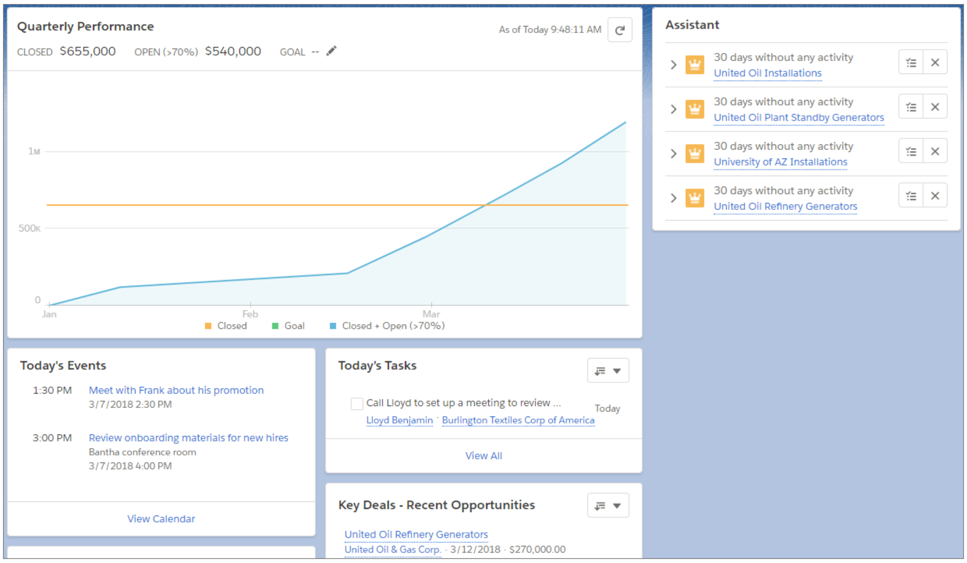
Let's assign this home page to the System Administrator profile so we can go look at it afterward.

1. Click **App and Profile**, then click **Assign to Apps and Profiles**.
2. Select the Sales app, then click **Next**.
3. Scroll down the list of profiles and select **System Administrator**, then click **Next**.
4. Review the assignment, and then click **Save**.

That's it. Now all users with the System Administrator profile see your New Home Page while working in the Sales app. Let's go have a look.

Look at Your Handiwork

Just for reference, here's an example of the standard default Home page. Think of this as the "before" in this Home page makeover.



30 days without any activity
United Oil Installations

30 days without any activity
United Oil Plant Standby Generators

30 days without any activity
University of AZ Installations

30 days without any activity
United Oil Refinery Generators

1:30 PM Meet with Frank about his promotion
3/7/2018 2:30 PM

3:00 PM Review onboarding materials for new hires
Bantha conference room
3/7/2018 4:00 PM

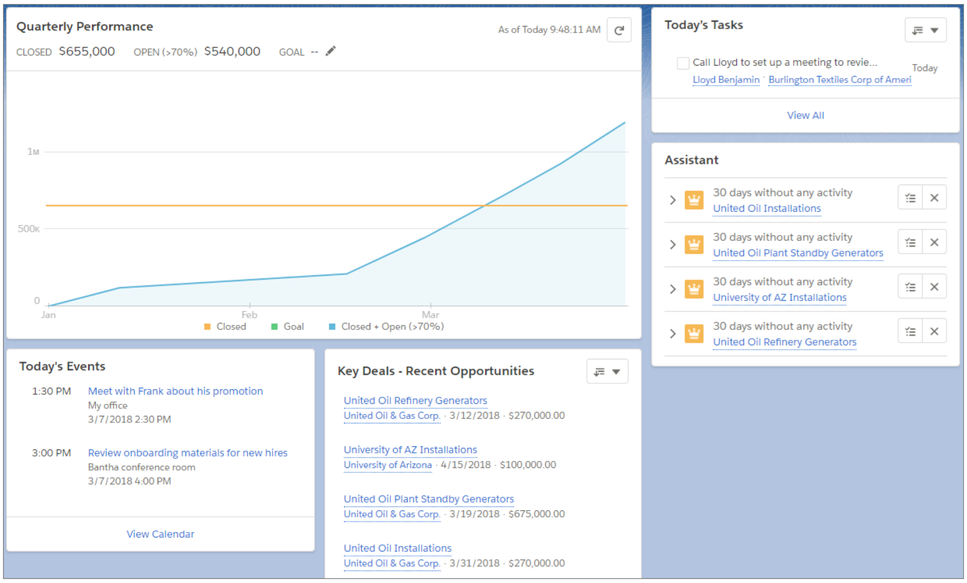
Call Lloyd to set up a meeting to review ... Today
Lloyd Benjamin · Burlington Textiles Corp of America

United Oil Refinery Generators
United Oil & Gas Corp. · 3/12/2018 · $270,000.00

Now, let's compare it to what we did.

1. Navigate to the Home page of the Sales app in Lightning Experience.

   Here's what it looks like now. It's not a whole lot different than before, but you can see that some components are in new places.


Quarterly Performance
CLOSED $655,000 OPEN (>70%) $540,000 GOAL --
As of Today 9:48:11 AM

Today's Tasks
Call Lloyd to set up a meeting to revie... Today
Lloyd Benjamin · Burlington Textiles Corp of Ameri
View All

Assistant
30 days without any activity
United Oil Installations
30 days without any activity
United Oil Plant Standby Generators
30 days without any activity
University of AZ Installations
30 days without any activity
United Oil Refinery Generators

Today's Events
1:30 PM Meet with Frank about his promotion
My office
3/7/2018 2:30 PM
3:00 PM Review onboarding materials for new hires
Bantha conference room
3/7/2018 4:00 PM
View Calendar

Key Deals - Recent Opportunities
United Oil Refinery Generators
United Oil & Gas Corp. · 3/12/2018 · $270,000.00
University of AZ Installations
University of Arizona · 4/15/2018 · $100,000.00
United Oil Plant Standby Generators
United Oil & Gas Corp. · 3/19/2018 · $675,000.00
United Oil Installations
United Oil & Gas Corp. · 3/31/2018 · $270,000.00

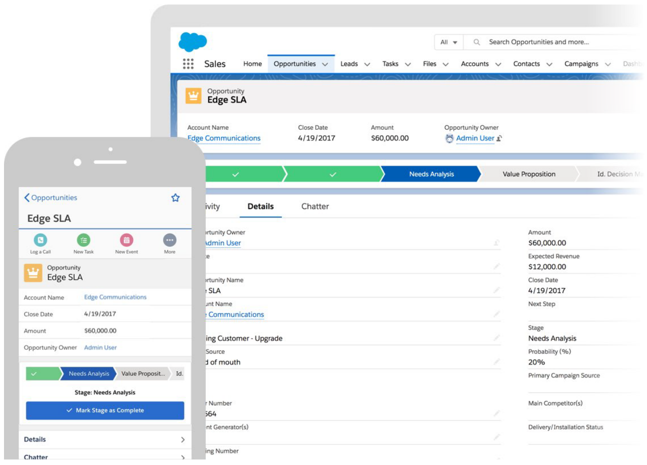
2. Bask in the glory of your newfound page customization skills.

All right! Let's move on to record pages.

CHAPTER 23 Those Lightning Experience Record Pages? You Can Customize Them, Too!

In this chapter ...

Use the Lightning App Builder to add, remove, or reorder components on a record page to give users a customized view for each object's records. Even cooler, you can customize a record page and assign it to specific Lightning apps to give your users access to a record page customized especially for the context of the app they're working in.



Just like the Home page, you can create a custom record page in different ways: Create it from scratch using a template, clone one of your other custom record pages, or edit an existing page. However, unlike the Home page, custom record pages are not only supported in Lightning Experience on desktop but also in the Salesforce mobile app. We're going to create a record page using a template, then check out how it looks on a desktop and on a phone.

Create a Custom Lightning Record Page

Let's build a custom opportunity record page from scratch.

We'll tweak the standard record page layout just a bit, so you can get a feel for how things go together. After you're comfortable with that, you can go to town and customize your record pages any way you like. Let's get started.

1. From Setup, enter `App Builder` in the Quick Find box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **Record Page** and start stepping through the wizard.
4. Name your page `New Opportunity Page`, and select **Opportunity**.

   Tip: Start typing an object's name in the Object field to filter the list and find what you're looking for more quickly.

5. Choose the **Header, Subheader, Right Sidebar** template, and click **Finish**.

   In the components pane, you see all the standard components available for opportunity record pages and any custom components that you've installed in your org.

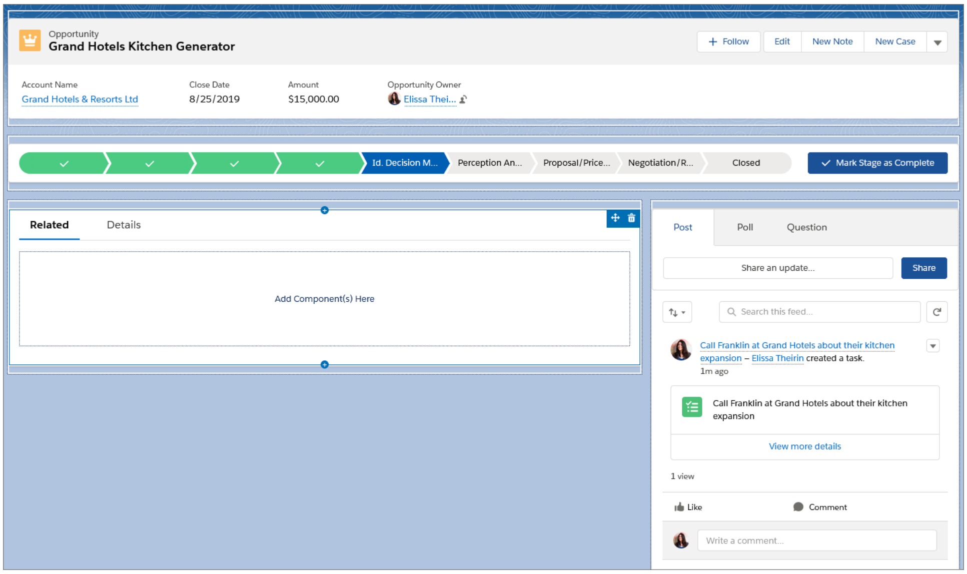
6. From the settings menu (⚙ ▾), select **Always show icons**.

   The icons that appear in the palette show what form factors each component supports. For example, if you add the Chatter Feed component to your page, it displays when you view the page on both a desktop and in the Salesforce mobile app. The same is not true for Chatter Publisher, which is supported on desktop only. We'll see this behavior in action when we test our finished page.

7. Drag the Highlights Panel component into the top region of the page.

   Click **See How It Works** in the component properties pane to find out where the highlights panel content comes from.

8. Add the Path component to the region below the highlights panel.
9. Add a Chatter component to the lower right region.
10. Add a Tabs component to the lower left region.

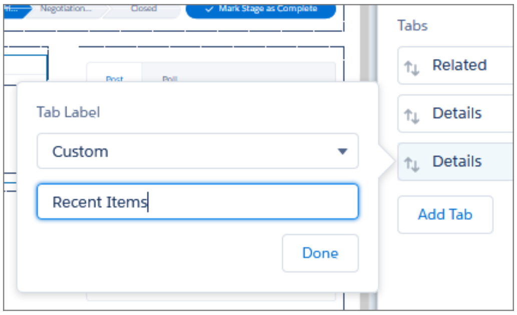


The Tabs component comes with some default tabs already in place. Let's add more.

11. In the Tabs component details pane, click **Add Tab**.

    By default, another Details tab is added. But because we already have one, let's change this new one to something else.

12. Click the second **Details** tab. From the Tab Label dropdown menu, select **Custom**, and give the tab a new label: `Recent Items`.



13. Click **Done**.

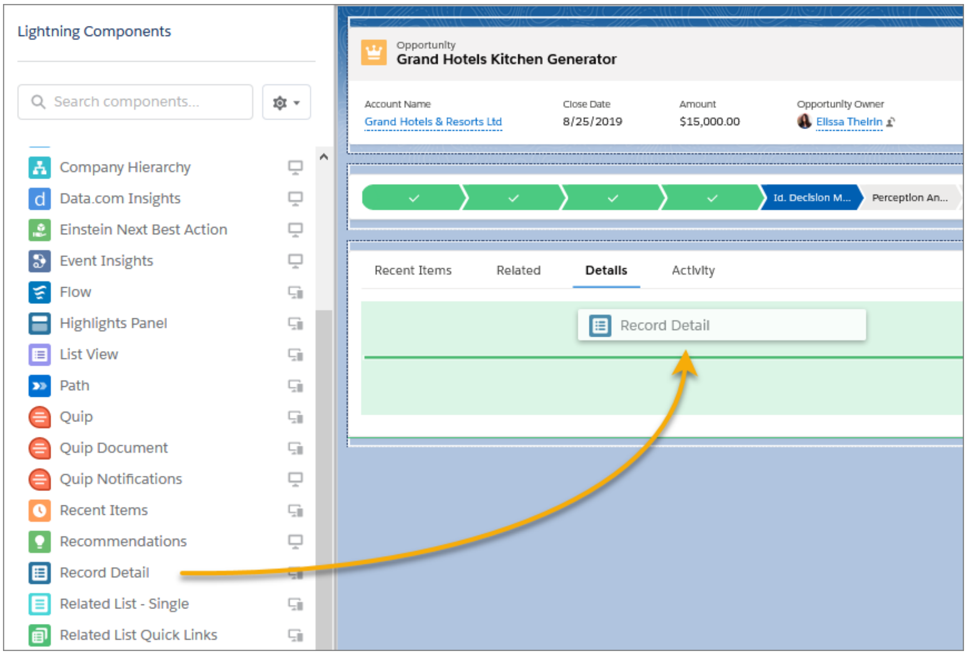
14. Create an Activity tab.

15. Drag the Recent Items tab to the top of the Tabs list in the properties pane.

The Recent Items tab is now in the first position in the tabs component. You can click around between the tabs, but nothing changes because the tabs don't have any components in them. They're empty. Let's fix that.

16. Select the Details tab.

17. Drag a Record Detail component right below the Details tab, into the green highlighted area.



18. Add a Related Lists component to the Related tab, an Activities component to the Activity tab, and the Recent Items component to the Recent Items tab.

19. Select the tabs component on the canvas, and in the properties pane, change the order of the tabs to: Details, Activity, Recent Items, then Related.

You can't drag the tabs inside the component to move them around. You can only adjust them in the properties pane.

20. Click **Save**, then **Not Yet**.

Make Your Record Page Dynamic

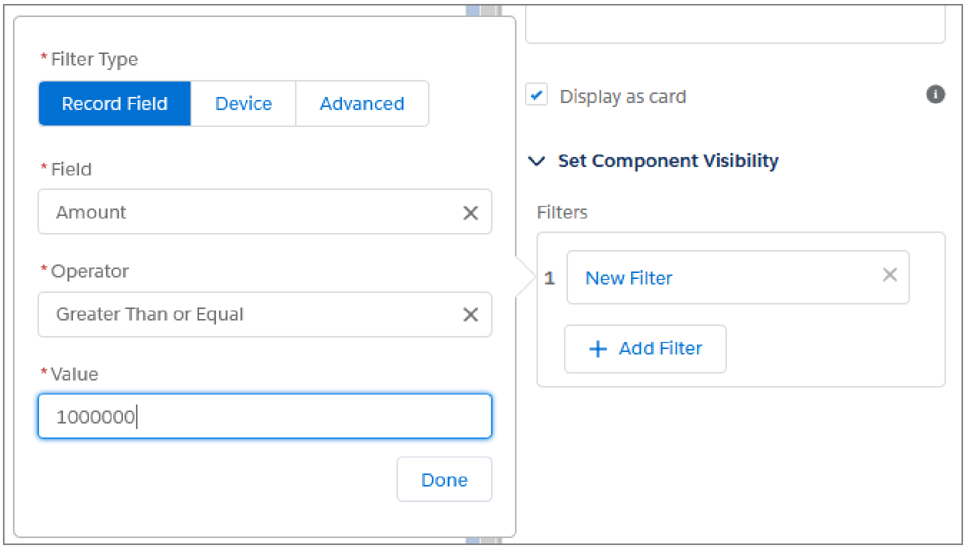
Did you know that you can control when a component appears on a Lightning record page? You can, by adding component visibility filter conditions and logic to its properties.

Component visibility properties appear when you select a component on a record, app, or Home page in the Lightning App Builder. This behavior applies to standard components, custom components, and components from AppExchange. No need to do anything to your custom components. It's all handled by the Lightning App Builder. If you don't define a filter, the component displays on the Lightning page as usual. When you define one or more filters and set the filter logic for a component, the component is hidden until the filter logic criteria are met.

Let's give it a shot. We'll construct filters to make a rich text component display when an opportunity's Amount is greater than or equal to $1 million, and its Stage is Closed Won.

1. Add a Rich Text component above the Chatter component on the page.
2. Enter this text in the component: *`A million dollar opportunity closed! Oh yeah!`*
3. In the component properties, make the text bold and centered, 18-point size, and change the font to something that appeals to you.
4. Keep **Display as card** selected.

   This setting makes the text inside the component more readable on Lightning pages by adding a white background to it instead of a transparent one. Toggle the setting off and back on to see what we mean.
5. Click **Add Filter**.
6. Set Field to **Amount**, if it's not already.
7. Set Operator to **Greater Than or Equal**. For Value, enter *`1000000`*.

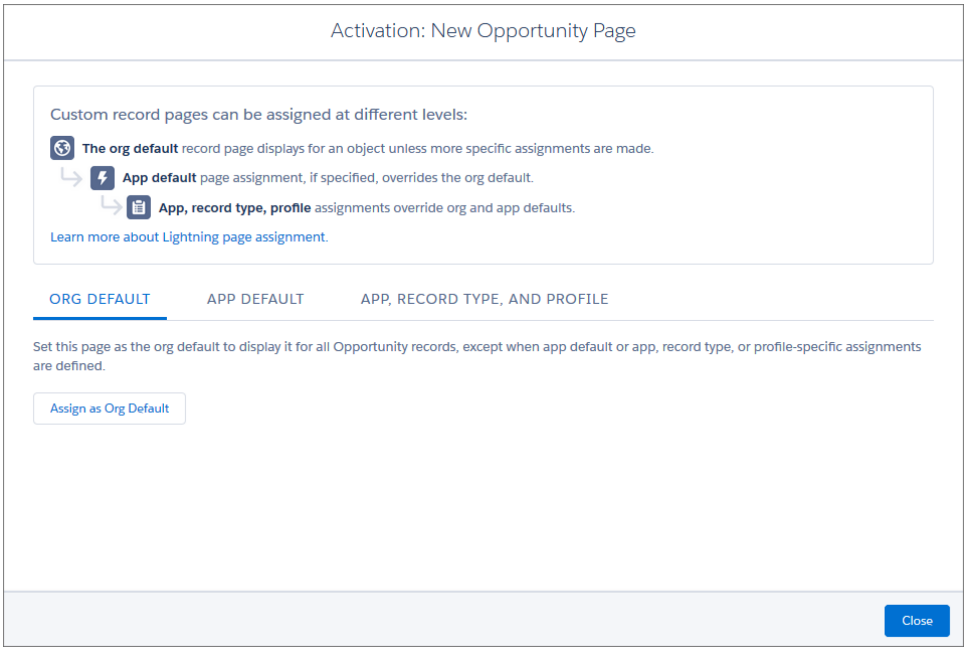


8. Click **Done**.
9. Click **Add Filter** again, and create another filter for the Stage field equal to Closed Won.

   Controlling whether a component displays based on field values isn't all you can do with visibility rules. As we mentioned, custom record pages are supported in Lightning Experience both on desktop and in the Salesforce mobile app. Visibility rules can also control whether components appear on a page based on the form factor (or device) you're viewing the page on. Let's set up rules for a component to appear only when the page is viewed on a phone.
10. Add another Rich Text component right below the first one.
11. Enter this text in the component: `This component is for mobile users only.`
12. Customize the text however you like, then click **Add Filter**.
13. Under Filter Type, click **Device**.
14. Set the Value field to Phone, then click **Done**.
15. Save the page, then click **Activation**.

Roll Out Your Custom Record Page to Your Users

It's time to spread the awesomeness! Let's activate the page. It's super easy.


Activation: New Opportunity Page

Custom record pages can be assigned at different levels:

The org default record page displays for an object unless more specific assignments are made.

App default page assignment, if specified, overrides the org default.

App, record type, profile assignments override org and app defaults.

Learn more about Lightning page assignment.

Set this page as the org default to display it for all Opportunity records, except when app default or app, record type, or profile-specific assignments are defined.

You have four options for activation.

- Make the page the org default for the object.
- Make the page the default object record page for specific Lightning apps.
- Assign the page to a combination of Lightning apps, record types, and profiles.
- Assign the page to a form factor, such as a desktop or phone.

Note the last bullet in the list. Not only can you create a page that is customized for the needs of your users, but you can go a step further and customize a page based on how your users access it. You can create uniquely customized record pages that only your mobile users see, containing only what they need while on the road or in the field. At the same time, desktop-only record pages can serve the needs of your users while they're working on their PCs or laptops.

Let's assign this page to a specific app, record type, and profile. We'll also make sure it's assigned to both the desktop and phone form factors, so we can view it in both places.

1. Click the App, Record Type, and Profile tab.
2. Click **Assign to Apps, Record Types, and Profiles**.
3. Assign the page to the Sales app, the Desktop and phone form factor, the Master record type, and the System Administrator profile.

4. Review the page assignments.
   The New Page column is populated with the name of the page we're activating: New Opportunity Page.
5. Click **Save**.

See? Easy peasy. Your customized record page is now live. Let's go check it out.

Tip: You may be thinking, "This is great, but what if I change my mind? How do I deactivate my custom page?" That's easy, too. Click **Activation**, click the App, Record Type, and Profile tab, and click **Remove Assignments**.

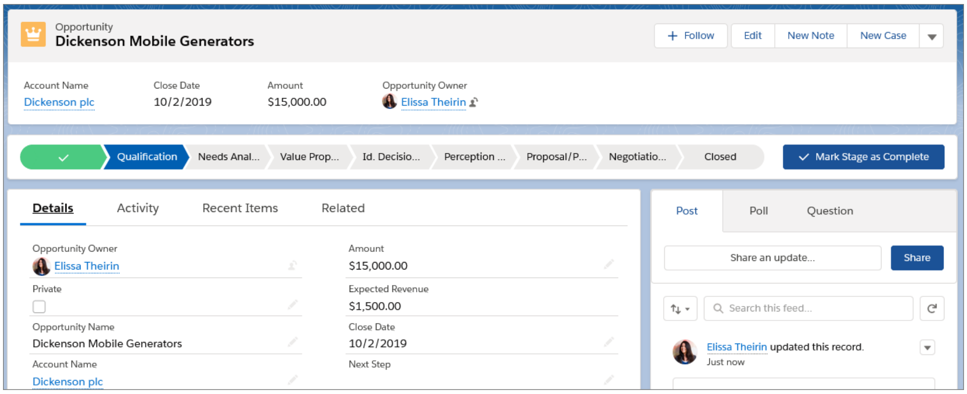
See What You Did There?

You've created a page and activated it. Now let's see it in action. First, we'll look at it on desktop.

1. Click **Back** in the App Builder header.
2. From the App Launcher (⁝⁝⁝), find and select **Sales**, then click the Opportunities tab.
3. Select any open opportunity with an amount below $1,000,000.

   You might have to refresh the opportunity page for the record page changes you just made to appear.

   Here's what the custom record page looks like, using the sample company, Dickenson Mobile Generators. Because you assigned the record page to the System Administrator profile, you can see it, but no other users in your org can. You can customize your different users' experiences by creating custom record pages and assigning them by app, record type, and user profile. Give your sales managers a different view of opportunities than your sales reps. Configure nonprofit account pages differently than standard business account pages.



Account Name: Dickenson plc
Close Date: 10/2/2019
Amount: $15,000.00
Opportunity Owner: Elissa Theirin

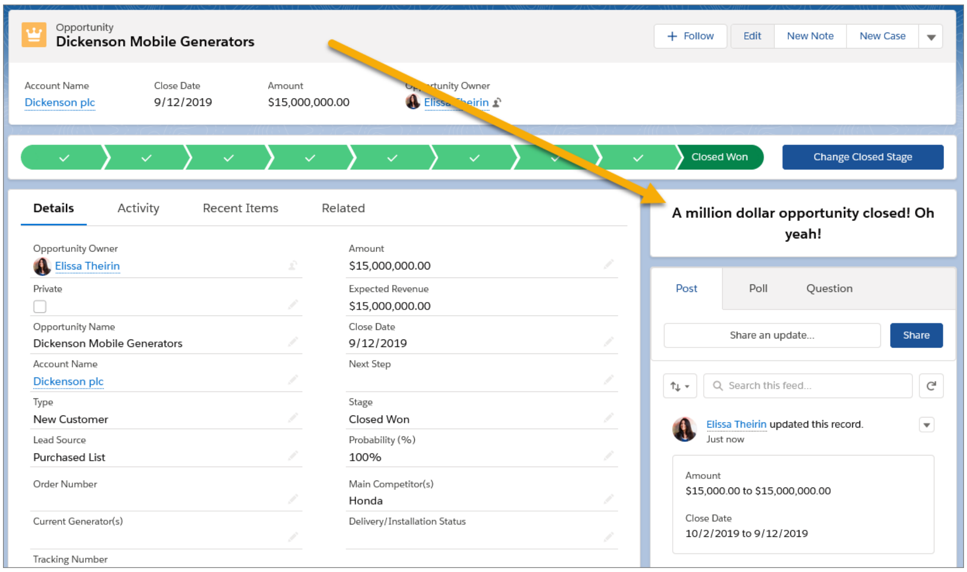
Opportunity Owner: Elissa Theirin
Amount: $15,000.00
Private
Expected Revenue: $1,500.00
Opportunity Name: Dickenson Mobile Generators
Close Date: 10/2/2019
Account Name: Dickenson plc
Next Step

Elissa Theirin updated this record.
Just now

Don't see the rich text components that you added to the page? That's for two reasons. For one component, it's because we're viewing the page on a desktop. We'll get to that shortly. For the other component, it's because the opportunity doesn't meet the criteria that you set. Let's change that.

4. From the page level actions in the Highlights panel, select **Edit**.
5. Change the opportunity's amount to be over $1,000,000, change the stage to Closed Won, then click **Save**.

   Woohoo, look at that! You didn't even have to refresh the page. When you save your changes and the filter criteria are met, the page automatically updates to show the rich text component and its message.



A million dollar opportunity closed! Oh yeah!

Opportunity Owner: Elissa Theirin	Amount: $15,000,000.00
Private: ☐	Expected Revenue: $15,000,000.00
Opportunity Name: Dickenson Mobile Generators	Close Date: 9/12/2019
Account Name: Dickenson plc	Next Step:
Type: New Customer	Stage: Closed Won
Lead Source: Purchased List	Probability (%): 100%
Order Number:	Main Competitor(s): Honda
Current Generator(s):	Delivery/Installation Status:
Tracking Number	

Elissa Theirin updated this record.
Just now

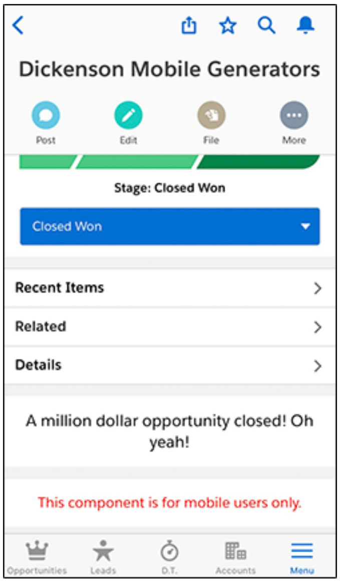
Amount
$15,000.00 to $15,000,000.00

Close Date
10/2/2019 to 9/12/2019

Now let's look at the page in the Salesforce mobile app.

6. Open the app on your phone.
7. If necessary, log in using your Trailhead Playground credentials.
8. Open the menu, tap the App Launcher, and open the Sales app.
9. Tap **Opportunities**, then navigate to the opportunity that you updated to Closed Won.

   What you see at first is what you'd expect: actions, record highlights, Path. But scroll down a bit...


Dickenson Mobile Generators

Stage: Closed Won

Closed Won

Recent Items

Related

Details

A million dollar opportunity closed! Oh yeah!

This component is for mobile users only.

The tabs in our Tabs component are stacked when viewed on a phone. You can tap them to drill in. But wait! There's one missing. Where's the Activity tab? Well, the Activities component isn't supported on a phone, so it was dropped from the page. And, since that caused the Activity tab to be empty, it was dropped from the page too.

Toward the bottom is the component that appeared when you updated the opportunity to be over a million dollars. But right below it is something we didn't see when we looked at the page on desktop: the mobile-only component.

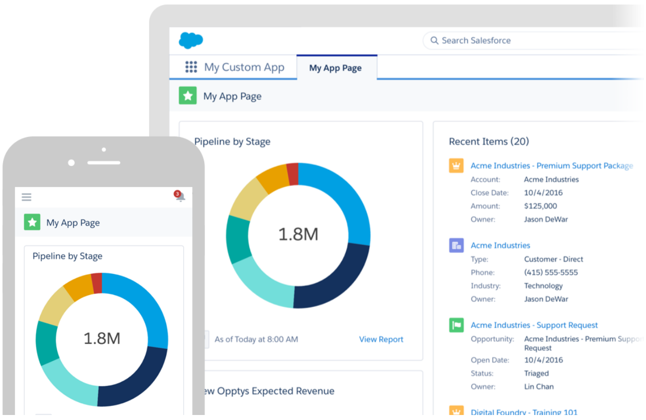
You did it! You've taken your first steps into a larger world.

Let's move on to app pages.

CHAPTER 24 The App Home Lightning Page

In this chapter ...

Add a custom home page for an app to the Salesforce mobile app and Lightning Experience app navigation bars to let your users easily access the objects and items that are most important in that app.



Acme Industries - Premium Support Package

Account:	Acme Industries
Close Date:	10/4/2016
Amount:	$125,000
Owner:	Jason DeWar

Acme Industries

Type:	Customer - Direct
Phone:	(415) 555-5555
Industry:	Technology
Owner:	Jason DeWar

Acme Industries - Support Request

Opportunity:	Acme Industries - Premium Supp Request
Open Date:	10/4/2016
Status:	Triaged
Owner:	Lin Chan

Digital Foundry - Training 101

Create an App Page

Let's build an app home page for a sales team.

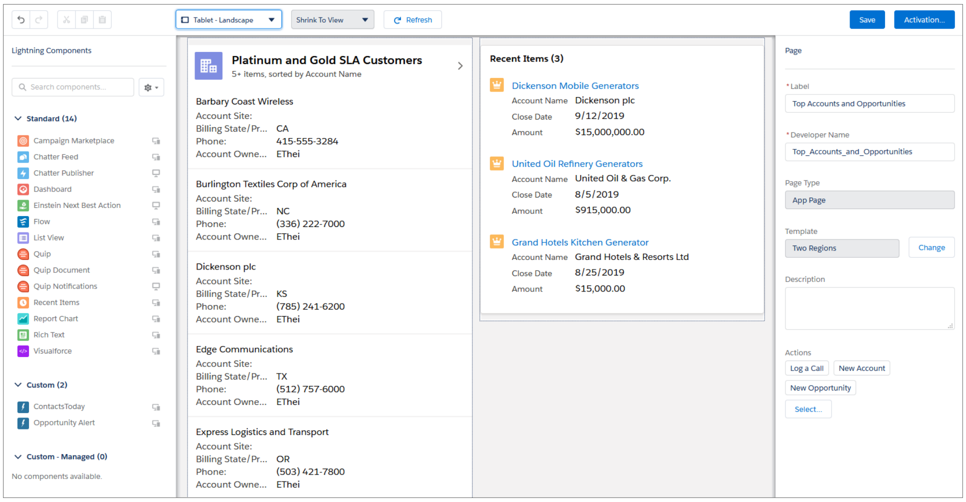
Your sales team needs to see top deals in the pipeline, with a visual interface that makes it easy to absorb key details at a glance. They want to see the most recent opportunities they've viewed and be able to drill into the record details with a single tap or click. And they want functionality to log calls and create accounts and opportunities on the go.

Let's get started!

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. Name your Lightning page `Top Accounts and Opportunities`.
5. Select the **Two Regions** template, and click **Finish**.

   If the Lightning App Builder walkthrough pops up, dismiss it.
6. Drag the List View component into the first region.
7. In the properties pane, under Filter, select the **Platinum and Gold SLA Customers** view, and set the number of records to display to 5.
8. Add a Recent Items component into the second region.
9. In the properties pane for the Recent Items component, click **Select** and add the Opportunity object to it.
10. In the properties pane, click **Page**, click **Select**, and then add these actions to the page.
    - Log a Call
    - New Account
    - New Opportunity

    You can add only global actions to a Lightning app page.
11. From the toolbar, change the view to Tablet-Landscape.
12. Click **Save**, and then **Not Yet**.


Tablet - Landscape | Shrink To View | Refresh | Save | Activation...

Lightning Components

Search components...

Standard (14)
- Campaign Marketplace
- Chatter Feed
- Chatter Publisher
- Dashboard
- Einstein Next Best Action
- Flow
- List View
- Quip
- Quip Document
- Quip Notifications
- Recent Items
- Report Chart
- Rich Text
- Visualforce

Custom (2)
- ContactsToday
- Opportunity Alert

Custom - Managed (0)

No components available.

Platinum and Gold SLA Customers
5+ items, sorted by Account Name

Barbary Coast Wireless
Account Site:
Billing State/Pr... CA
Phone: 415-555-3284
Account Owne... EThei

Burlington Textiles Corp of America
Account Site:
Billing State/Pr... NC
Phone: (336) 222-7000
Account Owne... EThei

Dickenson plc
Account Site:
Billing State/Pr... KS
Phone: (785) 241-6200
Account Owne... EThei

Edge Communications
Account Site:
Billing State/Pr... TX
Phone: (512) 757-6000
Account Owne... EThei

Express Logistics and Transport
Account Site:
Billing State/Pr... OR
Phone: (503) 421-7800
Account Owne... EThei

Recent Items (3)

Dickenson Mobile Generators
Account Name Dickenson plc
Close Date 9/12/2019
Amount $15,000,000.00

United Oil Refinery Generators
Account Name United Oil & Gas Corp.
Close Date 8/5/2019
Amount $915,000.00

Grand Hotels Kitchen Generator
Account Name Grand Hotels & Resorts Ltd
Close Date 8/25/2019
Amount $15,000.00

Page

* Label: Top Accounts and Opportunities

* Developer Name: Top_Accounts_and_Opportunities

Page Type: App Page

Template: Two Regions | Change

Description:

Actions: Log a Call | New Account | New Opportunity | Select...

Now that you've created your page, you're almost ready to add it to Lightning Experience and the mobile app.

Add Your App Page to Lightning Experience and the Salesforce Mobile App

Just like the other pages, your users can't access your app page until you activate it. During activation, you can customize the page's custom tab label, adjust its visibility, and set its position in the Salesforce mobile app and Lightning Experience app navigation bars, all in one place.

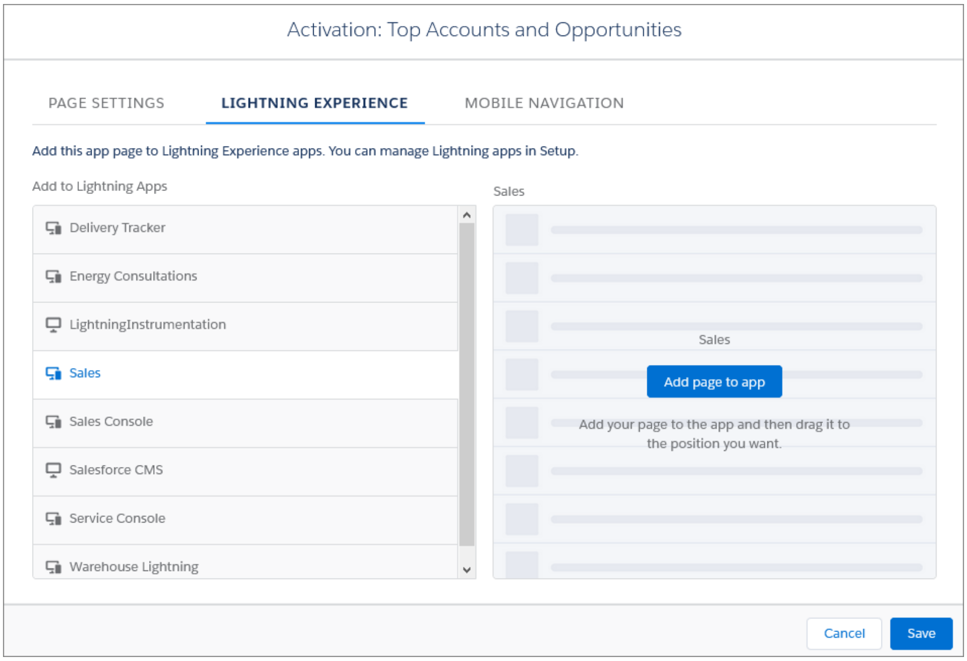
1. Click **Activation**.
2. Don't change the app name.

   By default, the label that you give the Lightning page is used as the label for its custom tab.
3. Change the icon to the blue lightning bolt icon.

   The icon that you choose here is used as the icon for the page in the mobile app and in Lightning Experience.
4. Keep the tab's visibility open to all users.

   Tip: The `Activate for the System Administrators only` setting is useful while you're working on your Lightning page. Restricting your page to administrators only means that you can see and test the page, but your users can't see it until you're ready to expose it to them.

5. Click the Lightning Experience tab.
6. Select the Sales app, and click **Add page to app**.


Activation: Top Accounts and Opportunities

Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.

Add to Lightning Apps

- Delivery Tracker
- Energy Consultations
- LightningInstrumentation
- Sales
- Sales Console
- Salesforce CMS
- Service Console
- Warehouse Lightning

Sales

Sales

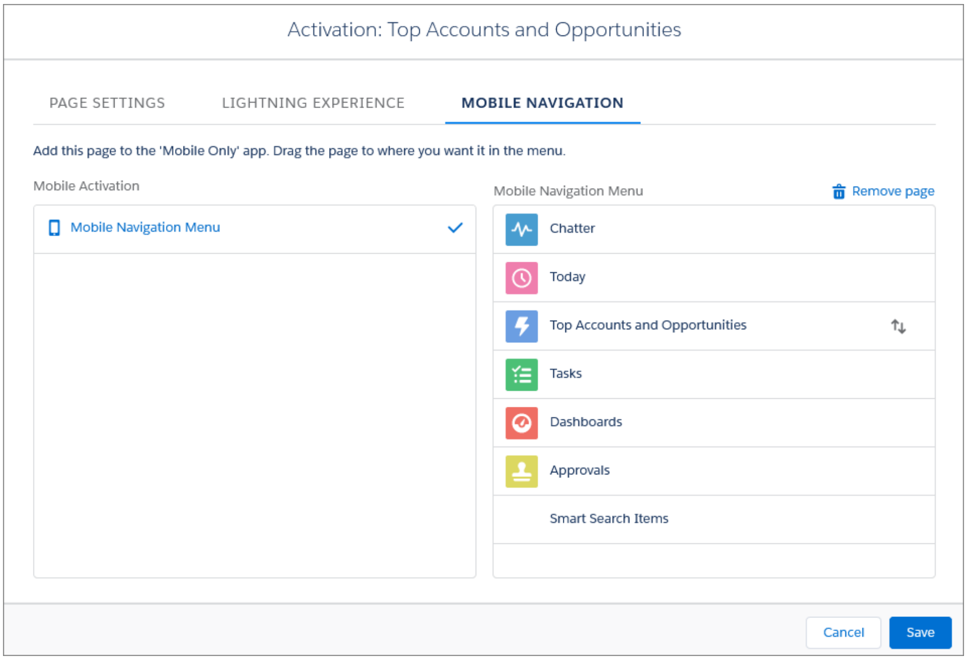
Add your page to the app and then drag it to the position you want.

The page you're adding to the menu appears in the second position by default. Let's leave it there. If you put it into the top position, it becomes the landing page for all your Lightning Experience users.

7. Click the **Mobile Navigation** tab.
8. Select **Mobile Navigation Menu**, and click **Add page to app**.
   This adds the app page to the mobile navigation in the "Mobile Only" Lightning app in the Salesforce mobile app. Adding the page to the Sales app in Lightning Experience like we did previously, ensures that the app page appears in the Sales app within the Salesforce mobile app as well.

   By default, new pages you add to the navigation in the "Mobile Only" app appear below the Smart Search Items menu item. If you leave the Top Accounts and Opportunities page there, it will appear in the Apps section of the menu. We don't want that, so let's move it up.

9. Drag the page to below the Today menu item.



Activation: Top Accounts and Opportunities

Add this page to the 'Mobile Only' app. Drag the page to where you want it in the menu.

10. Click **Save**.

Your app home page is now ready for your mobile and Lightning Experience users!

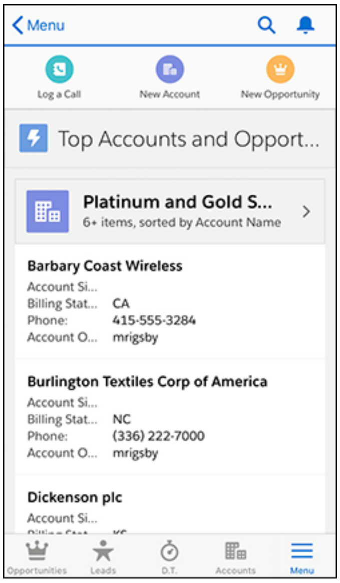
Test Your App Page in the Salesforce Mobile App

Now let's look at it on your phone.

1. Open the Salesforce mobile app.
2. Log in using your Trailhead Playground credentials. (If you don't know your Trailhead Playground credentials, see this article.
3. From the menu, tap the App Launcher, and open the Sales app.

   You might have to pull and refresh the Sales app menu to see the changes.
4. Tap **Top Accounts and Opportunities**.

Here's your Lightning page! It has the lightning bolt icon you chose, and the three actions that you added are in the action bar.



Barbary Coast Wireless
Account Si...
Billing Stat... CA
Phone: 415-555-3284
Account O... mrigsby

Burlington Textiles Corp of America
Account Si...
Billing Stat... NC
Phone: (336) 222-7000
Account O... mrigsby

Dickenson plc
Account Si...

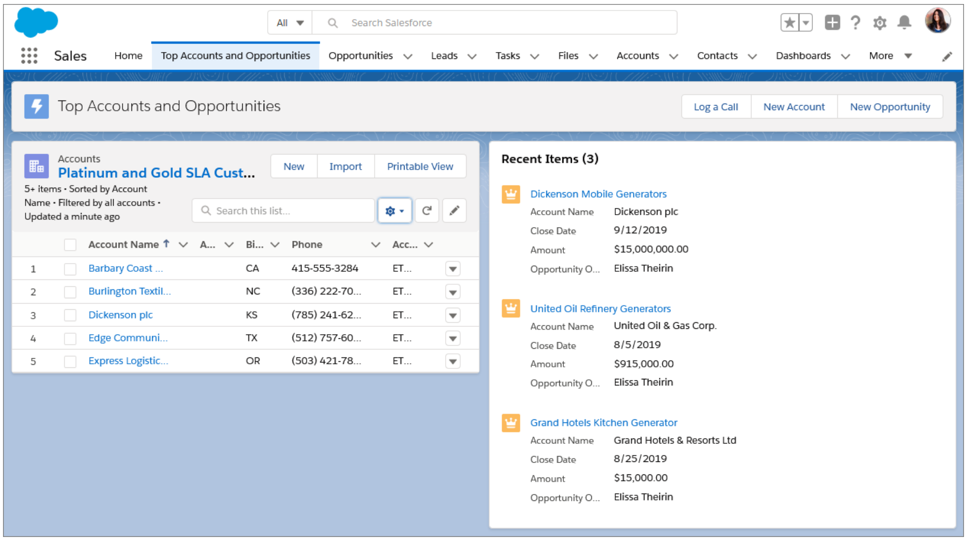
5. Play around with your page. Scroll up and down to see the components, and tap an action icon to see what happens.

Test Your App Page in Lightning Experience

You've created your page and activated it. Now let's see it in action!

1. If you're still in the App Builder, click **Back** to return to Setup.
2. From the App Launcher (⋮⋮⋮), find and select **Sales**.
3. Click **Top Accounts and Opportunities** from the app navigation bar.

   Here's your Lightning page! The lightning bolt icon appears here too, and the three actions that you added are in the highlights panel.



5+ items • Sorted by Account Name • Filtered by all accounts • Updated a minute ago

	Account Name ↑	A...	Bi...	Phone	Acc...
1	Barbary Coast ...		CA	415-555-3284	ET...
2	Burlington Textil...		NC	(336) 222-70...	ET...
3	Dickenson plc		KS	(785) 241-62...	ET...
4	Edge Communi...		TX	(512) 757-60...	ET...
5	Express Logistic...		OR	(503) 421-78...	ET...

Dickenson Mobile Generators

Account Name	Dickenson plc
Close Date	9/12/2019
Amount	$15,000,000.00
Opportunity O...	Elissa Theirin

United Oil Refinery Generators

Account Name	United Oil & Gas Corp.
Close Date	8/5/2019
Amount	$915,000.00
Opportunity O...	Elissa Theirin

Grand Hotels Kitchen Generator

Account Name	Grand Hotels & Resorts Ltd
Close Date	8/25/2019
Amount	$15,000.00
Opportunity O...	Elissa Theirin

CHAPTER 25 Custom Lightning Components in the Lightning App Builder

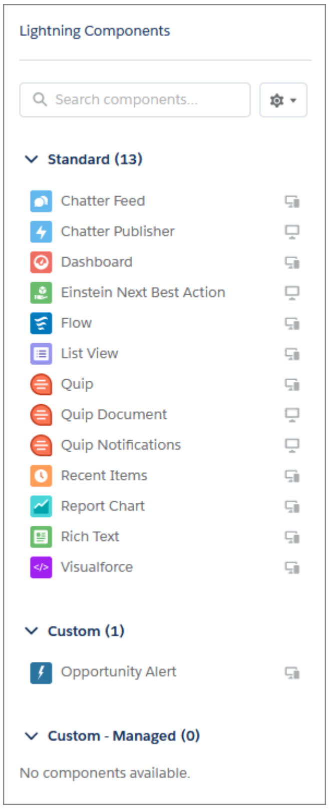
In this chapter ...

Creating a custom Lightning component requires programming skills. Or, you can install an existing Lightning component from the AppExchange.

Important: Point-and-click admins, you can do this unit! We're providing you with a custom component that's already configured. So don't worry. You don't need programming skills to earn this badge.

You can create a custom Lightning component using two programming models, Lightning Web Components and Aura Components. See the modules Aura Components Basics and Lightning Web Components Basics.

Custom components in your org that are configured for use in the Lightning App Builder appear in the Lightning Components pane.



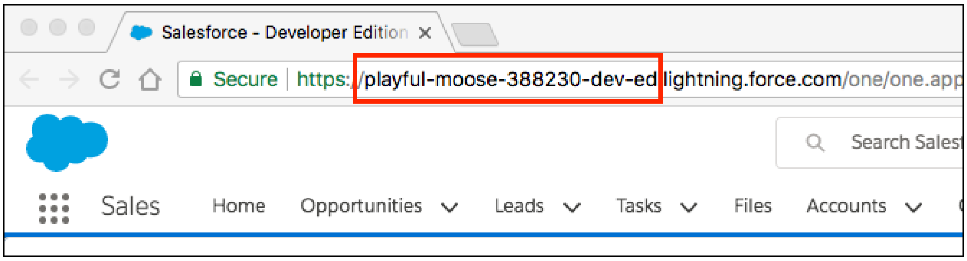
Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, you need to:

1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.
2. Deploy MyDomain in your org. When you deploy MyDomain , references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

The custom component we're providing for this module has already been configured for the Lightning App Builder.

My Domain Is Already On in Your Trailhead Playground

Do not attempt to turn on My Domain, or change its settings, in your Trailhead Playground. By default, My Domain is already active in every Trailhead Playground.


Salesforce - Developer Edition
Secure | https://playful-moose-388230-dev-ed.lightning.force.com/one/one.app
Search Sales
Sales
Home
Opportunities
Leads
Tasks
Files
Accounts

In your production org, My Domain lets you create a subdomain unique to your organization. With My Domain, you replace the instance URL that Salesforce assigns you, such as `https://na17.lightning.force.com`, with your chosen subdomain, for example, `https://mydomainname.lightning.force.com`.

My Domain is required to create custom Lightning components and set up single sign-on (SSO) in an org. To learn more about My Domain, check out this knowledge article. To learn how to activate it in your production org, see the User Authentication module.

Install a Custom Lightning Component

We've provided a custom Opportunity Alert Lightning component that you can add to your Lightning page. Let's install it into your org.

Launch your Trailhead Playground by going to any unit with a hands-on challenge, scrolling to the bottom of the page, and clicking **Launch**. If you see a tab in your org labeled Install a Package, great! Follow the steps below.

If not, click the App Launcher icon to launch the App Launcher, then click **Playground Starter** and follow the steps. If you don't see the Playground Starter app, copy this package installation link and check out Install a Package or App to Complete a Trailhead Challenge on Trailhead Help.

1. Click the **Install a Package** tab.
2. Paste *`04t2E000000161fSQAQ`* into the field.
3. Click **Install**.

4. Select **Install for Admins Only**, then click **Install**.

After the installation completes, it's time to add the component to your page.

Add the Custom Lightning Component to Your App Page

1. From Setup, enter *`App Builder`* into the Quick Find box, and then click **Lightning App Builder**.
2. Click **Edit** next to the Top Accounts and Opportunities page.
3. Add the Opportunity Alert component above the List View component.
4. In the properties pane, change `Days Since Last Modified` to *6*.

   Did you notice that the text inside the component changed to match the new field value?
5. Deselect **Has Open Tasks**, then click **Save**.

   Tip: Because you activated the page previously, you don't have to activate it again.

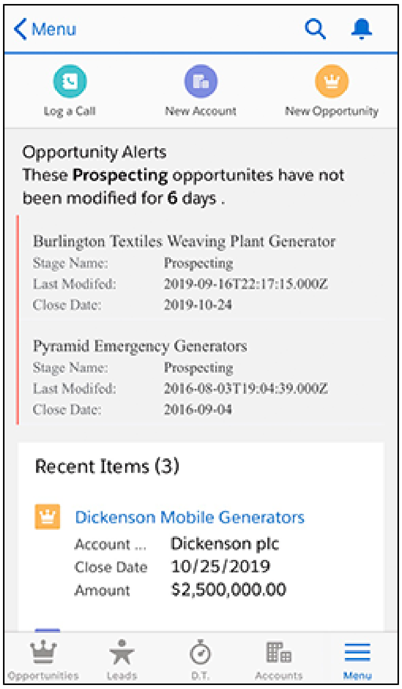
All right! The changes you made are immediately available in Lightning Experience and the mobile app. Let's see what the page looks like now.

Test the Custom Component

Let's look at it in the Salesforce mobile app first.

1. Open the app on your mobile device.
2. If necessary, log in using your Trailhead Playground credentials.
3. Go to the Top Accounts and Opportunities page in the Sales app and scroll down until you see the new component.

   You might have to refresh the screen to see your changes.



Opportunity Alerts
These **Prospecting** opportunites have not been modified for **6** days .

Burlington Textiles Weaving Plant Generator

Stage Name:	Prospecting
Last Modifed:	2019-09-16T22:17:15.000Z
Close Date:	2019-10-24

Pyramid Emergency Generators

Stage Name:	Prospecting
Last Modifed:	2016-08-03T19:04:39.000Z
Close Date:	2016-09-04

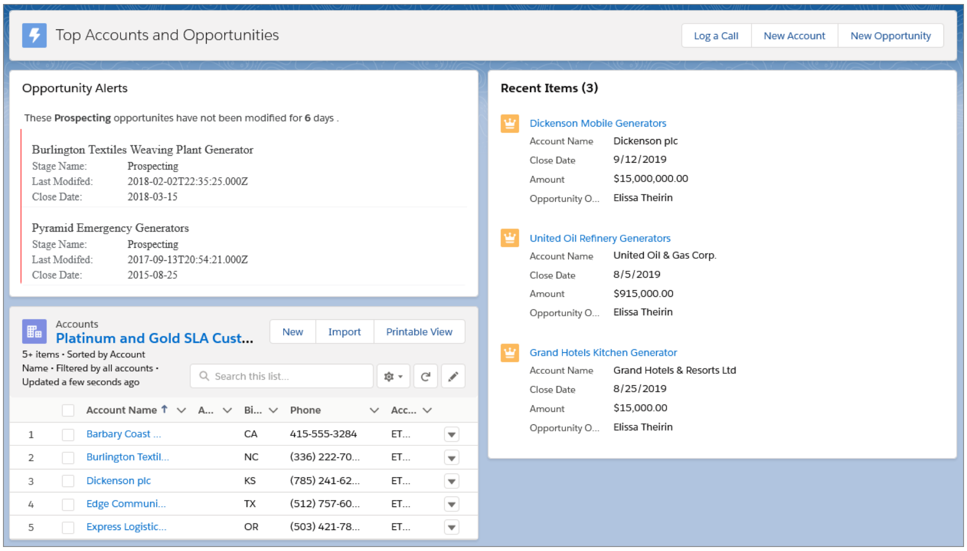
Recent Items (3)

Dickenson Mobile Generators

Account ...	Dickenson plc
Close Date	10/25/2019
Amount	$2,500,000.00

4. Go back to your org and view the Top Accounts and Opportunities page in Lightning Experience.

   Because you created the page using the Two Columns template, it uses the two-column format when you view it on the desktop or a tablet.



These **Prospecting** opportunites have not been modified for **6** days .

Burlington Textiles Weaving Plant Generator
Stage Name: Prospecting
Last Modifed: 2018-02-02T22:35:25.000Z
Close Date: 2018-03-15

Pyramid Emergency Generators
Stage Name: Prospecting
Last Modifed: 2017-09-13T20:54:21.000Z
Close Date: 2015-08-25

5+ items • Sorted by Account Name • Filtered by all accounts • Updated a few seconds ago

	Account Name	A...	Bi...	Phone	Acc...
1	Barbary Coast ...		CA	415-555-3284	ET...
2	Burlington Textil...		NC	(336) 222-70...	ET...
3	Dickenson plc		KS	(785) 241-62...	ET...
4	Edge Communi...		TX	(512) 757-60...	ET...
5	Express Logistic...		OR	(503) 421-78...	ET...

Dickenson Mobile Generators
Account Name: Dickenson plc
Close Date: 9/12/2019
Amount: $15,000,000.00
Opportunity O...: Elissa Theirin

United Oil Refinery Generators
Account Name: United Oil & Gas Corp.
Close Date: 8/5/2019
Amount: $915,000.00
Opportunity O...: Elissa Theirin

Grand Hotels Kitchen Generator
Account Name: Grand Hotels & Resorts Ltd
Close Date: 8/25/2019
Amount: $15,000.00
Opportunity O...: Elissa Theirin

Congratulations! You used the Lightning App Builder to create an app page, personalize your Lightning Experience Home page, and customize a Lightning Experience record page.

You've got the skills now to make the Salesforce mobile app and Lightning Experience handier and more powerful for your users. Go forth and customize!

LIGHTNING EXPERIENCE DEVELOPMENT

CHAPTER 26 User Interface Development Considerations

In this chapter ...

Lightning Experience is a brand new, amazing opportunity for developers on the Lightning Platform platform. In this module we're going to look at many of the aspects that make Lightning Experience different for developers. Before we get to that, though, we want to have a quick chat about a couple higher-level concerns.

Let's start with something we want to be completely honest with you about: Lightning Experience isn't finished yet. If you've visited some of the other Lightning Experience Trailhead modules, you already know there is plenty of work for us to do just building out the basic Salesforce application. On the Lightning Platform side of things, we have plenty to do as well. Most things are working great, but there are still some things we haven't gotten to yet.

About this we want you to know two things: First, we're very hard at work turning Lightning Experience into a complete development platform for your apps. And—Safe Harbor, yadda yadda yadda—you'll be getting important missing features in the next release, and the next, and so on.

Second, we very much want your feedback. Tell us what's working great to make us feel good, and tell us about what's not, to keep us working hard, to shape priorities, and to help us make sure Lightning Experience becomes the best cloud-based development platform you can imagine.

Finally, we want to answer some really important questions that you're asking us about where you should put your efforts as you grow your existing apps and build new ones. Put simply, what's the future of user interface development and custom apps on the Lightning Platform Platform?

We're glad you asked! We're so excited about the answer, we're going to take the rest of this unit to go beyond just answering the question, and try to give you insights into our thinking.

Raising the Bar for Web App User Interfaces

Over the last decade or more, we've all seen the bar continually be raised for user experience in web applications. Users expect responsive, fully featured, and highly interactive, immersive experiences literally at their fingertips.

We first saw this in single-purpose apps. Services like Google Maps introduced direct manipulation of user interface elements. Analytics applications brought dynamic, interactive chart drill-downs. Even the humble sign-up or log in form comes with dynamic error feedback when users enter invalid data, animations, transitions, and more. Interactivity is no longer a novelty, it's the norm.

And the scale has grown. The expectation for individual components has quickly spread to the core application experience. Today web applications feature things like sliding menus, animated page transitions, and dynamic master-details. There's also app-style elements such as overlays and modal windows. The difference between native applications and web applications has never been smaller.

So what does this mean for Salesforce?

The traditional Salesforce experience, lovingly known as Salesforce Classic, is an example of a **page-centric** web application model. It's great for basic functionality, but it's challenging to deliver the new, more dynamic experience that users expect. Fundamentally, this is because it relies on the server to generate a new page every time you interact with the application.

To deliver a more interactive experience, you need help from JavaScript on the client-side. In this new **app-centric** model, JavaScript is used to create, modify, transform, and animate the user interface rather than completely replacing it a page at a time. This model is exciting, interactive, and fluid. This is the new Lightning Experience.

Both the page-centric and app-centric models are here to stay. A quick look around the web is enough to show that most web applications take advantage of both approaches. Combining these models lets applications deliver the right type of experience for the right use case.

Let's take some time to explore the different options that the Salesforce Platform offers for these models.

Classic Visualforce

Visualforce is a robust, mature platform for building page-centric apps. The Visualforce framework provides a robust set of tags that are resolved on the server, and that work alongside standard or custom controllers to make database and other operations simple to implement.

Let's review some basics.

UI Generation
    Server-side

Data and Business Logic

Apex standard or custom controller

Workflow

1. User requests a page
2. The server executes the page's underlying code and sends the resulting HTML to the browser
3. The browser displays the HTML
4. When the user interacts with the page, return to step one

Pros

- Tried and true model
- Easy to implement for greater productivity
- Naturally splits large applications into small, manageable pages
- Has built-in metadata integration

Caveats

- Limited interactivity (aside from added JavaScript functionality)
- Higher latency, which degrades mobile performance

Visualforce is conceptually similar to other page-centric technologies like PHP, ASP, JSP, and Ruby on Rails. Salesforce's rich metadata infrastructure makes Visualforce a productive solution. The standard controller makes it easy to access objects directly and via relationships without executing a single query. Other metadata-aware components are similarly plug-and-play: add markup to a page and you're done. These capabilities are alive and well on the platform, and they're still suitable for many use cases.

Visualforce as a JavaScript Application Container

When you think about it, Visualforce pages are just HTML pages with extra tags resolved by the server. As a result, you can use an empty Visualforce page as a container for a JavaScript application. In this scenario, you don't use Visualforce tags to build your user interface. Instead, you load your JavaScript application in an empty page. Then the user interface is generated on the client-side by the JavaScript application. These applications are generally referred to as *single-page applications*, or SPAs, and are often built using third-party frameworks like AngularJS or React.

Let's review some basics.

UI Generation

Client-side (mostly JavaScript)

Data and Business Logic

Remote Objects or JavaScript Remoting, Apex controller

Workflow

1. The user requests the “empty” Visualforce page containing a page skeleton and JavaScript includes
2. The page is returned to the browser
3. The browser loads the JavaScript application
4. The JavaScript application generates the UI
5. When a user interacts with the application, the JavaScript modifies the user interface as needed (return to the previous step)

Pros

- Enables highly interactive and immersive user experiences

Caveats

- Complex
- No built-in metadata integration
- Lack of an integrated developer experience. The Lightning Platform Developer Console doesn’t explicitly support these JavaScript applications. Typically, you have to load them as static resources, and that’s a cumbersome experience.

We want to be clear. If you can live with the caveats we’ve described, this is a perfectly good way to build interactive applications today. It’s the reason we originally built toolkits like Remote Objects and JavaScript remoting. If you’re a confident AngularJS or React or other JavaScript framework developer, your expertise will take you a long way developing apps for Salesforce with the tools you know.

But, if you’re open to new things, we think we have some great ideas for what the next level is in web-based application development.

Lightning Components

Lightning components are part of the Salesforce user interface framework for developing dynamic web applications for desktop and mobile devices.

Note: As of the Spring ’19 release (API version 45.0), you can build Lightning components using two programming models: the Lightning Web Components model and the original Aura Components model. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page.

Let's take a look at an overview.

UI Generation

Client-side (JavaScript)

Data and Business Logic

Lightning Data Service, Apex

Workflow

1. The user requests an application or a component
2. The application or component bundle is returned to the client
3. The browser loads the bundle
4. The JavaScript application generates the UI
5. When the user interacts with the page, the JavaScript application modifies the user interface as needed (return to previous step)

Pros

- Enables highly interactive and immersive user experiences
- Aligns with Salesforce user interface strategy
- Built on metadata from the foundation, accelerating development

Choosing the Right Tool for the Job

Visualforce has been around for a while, it's a mature, well-understood platform for building your apps. It's not going away. Lightning Components is the new kid on the block. It's got a lot going for it, but, well, you know. It's a stranger to you right now.

Here's the thing: You don't have to choose one or the other.

Think of the page-centric and app-centric models as tools in your development tool belt—one isn't always better than the other, and you'll get the most out of them if you understand their strengths and their trade-offs. Use the right tool for the job at hand.

Here are some guidelines to help you decide—but remember, you're The Decider. In the end, use the tool that feels right *to you*. Also, keep in mind that tools evolve. These guidelines will evolve too.

Job	Recommendation
I'm Developing for Lightning Experience	We highly recommend Lightning Components. Lightning Experience was *built* with Lightning Components, and the two fit together like hand and glove. You can certainly use Visualforce if you have existing code or a project underway. Visualforce for Lightning Experience is fully supported, with a few constraints. But you won't find a better tool for Lightning Experience than working in its native language, Lightning Components.
I'm Developing for the Salesforce Mobile Application	We recommend Lightning Components. Visualforce characteristics, especially the page-centric orientation, can be a poor match for mobile apps with limited, high-latency network connections and limited compute resources. Lightning Components, by contrast, was designed *specifically* to handle this context. Both Visualforce and Lightning Components use similar tag-based markup. For example, Visualforce markup for an input field is `<apex:inputText>` while for Lightning Components it's `<lightning:input>`. So what's the difference? Well, Visualforce processes markup tags on the Salesforce server. Lightning Components process markup on the client. The advantage of processing on the client is that the HTML block for the entire page isn't passed back and forth between the client and the server with every interaction. With a few exceptions, Lightning Components are better for Salesforce mobile development. Some cases call for Visualforce as a JavaScript application. See the Lightning Components Developer's Guide for more information.
I'm Building a Page-Centric Experience with Limited Client-Side Logic	Use Visualforce pages to ensure development velocity and manageability.
I'm Building an Interactive Experience with JavaScript to Meet User Experience Requirements	Use Lightning Components but refer to limitations documentation first.

Job	Recommendation
I'm Committed to a JavaScript Framework Such as AngularJS or React	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
I'm Enabling Non-Developers to Build Apps by Assembling Standard or Custom Components	Use Lightning Components to create custom components that can be used in the Lightning App Builder.
I'm Building an Interactive Experience with JavaScript and I Need a Third-Party Framework	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
I'm Adding User Interface Elements	For example, say you want to add a tab to a record home. This task is a simple drag-and-drop in the Lightning App Builder. Use Lightning Components to create custom user interface elements.
I'm Building a Community for Customers	Use Experience Builder to create a Lightning-based community leveraging Lightning Components.
I'm Building a Community for Partners	Use Experience Builder to create a Lightning-based community leveraging Lightning Components.
I'm Exposing a Public-Facing Unauthenticated Website	Use Experience Builder to create a Lightning-based site leveraging Lightning Components.
I'm Rendering Pages as PDFs in My Application	Use Visualforce. Lightning Components don't support rendering as PDF output yet.
I'm Adding to an Existing Project with Lots of Visualforce Pages	Continue to use Visualforce. Consider opportunistically moving elements to Lightning Components using Lightning Components for Visualforce.
I'm Committed to Investing in the Latest Technology	You're right there with us! Dive in to Lightning Components.
I'm Starting a Brand New Project	Use Lightning Components. If you're not familiar with them, there's no better time than now to learn!
I'm Building a Flow to Guide Users Through a Business Process	Use Lightning Components to customize the functionality and look-and-feel of the flow's screens.

Choosing the Right Tool for Your Organization

When you think about choosing a tool, it's important to focus on more than just the job at hand. You also want to consider your organization as a whole and your role within your organization. Let's look at how some different roles can best leverage Salesforce's development models.

Role	Recommendation
ISV Partner	Start using Lightning components for new apps or new features in existing apps. Package these units for subscriber use in both Salesforce Classic and Lightning Experience.
SI	Start using Lightning components for new implementations. For in-progress implementations, continue using Visualforce.
Professional developers who are JavaScript gurus and experienced with Visualforce	Continue using Visualforce with your preferred JavaScript framework. Explore Lightning components and consider switching down the line.
Citizen developers who use standard Visualforce components for pages	Continue using Visualforce, but consider checking out the Lightning App Builder.
Point-and-click admins	Use the Lightning App Builder to create apps and customizations. Buddy up with developers and partners to build custom Lightning components.

Migrating to Lightning Components

Here's the good news. Despite the shift to Lightning Experience and an increased focus on Lightning components, your Visualforce pages still work on Salesforce. It doesn't matter if you're using the new interface or your old friend Salesforce Classic—Visualforce is able to work with both. You don't have to convert any existing Visualforce pages to keep using them for a long time.

But, because web applications are taking more advantage of the app-centric model, we encourage all Salesforce developers to learn at least the basics of Lightning components. You'll want to use these components in your future development work.

Now is the perfect time for you to take the first steps. Features such as the Lightning App Builder and Lightning Components for Visualforce let you "dip your toe in the water," and try using as little as a single

Lightning component in a new or existing page. You can use these embedded components in both Lightning Experience and Salesforce Classic. It's so easy, you'd be nuts not to give it a try.

We know that migrating to a new development framework is daunting. But we're here for you. This trail is loaded with all the tips, tricks, and gotchas that you need to successfully adopt Lightning Experience development.

CHAPTER 27 Visualforce and Lightning Experience

In this chapter ...

Lightning Experience is a whole new world, and we hope you think it's exciting. Behind the Lightning Experience user interface is a new way of delivering the Salesforce application, one that brings significant changes to the way Visualforce apps run. For the most part, your Visualforce apps should "just work," but there *are* some things you should know before you make the jump to Lightning Experience.

The technical details of how Lightning Experience is built and how it runs Visualforce apps are really cool, and important for actual development work. When you're ready for those details, the Visualforce & Lightning Experience module will show you the way. But here we're going to stay at a high level, and divide things up into what works and what doesn't, what you'll want to update, and other issues that will help you plan your Lightning Experience migration development effort.

What Works

The list of what parts of Visualforce work in Lightning Experience is pretty long. It's very nearly as long as the entire Visualforce features list. So before we get to the things that *aren't* on the list, let's think positive, and check off some of the many things you can count on.

First of all, the fundamental mechanisms of how Visualforce works remain the same. Whether your pages use the standard controller, custom controllers, JavaScript remoting, or Remote Objects, your connection to Salesforce works the same.

Note: If you're using JavaScript extensively, or if you're using other APIs to access Salesforce, you might have some work to do. We'll get to that.

Second, Visualforce markup remains the same. There are a few tags and attributes that behave differently in Lightning Experience, and a very few that we recommend against using or that just don't work. But otherwise the way you write code for Visualforce pages and components is unchanged.

Third, most of the ways you can use Visualforce to customize your organization work just fine in Lightning Experience—though as you can no doubt imagine, with an all new user interface, these customizations have moved to different places.

Let's dive just a bit into the specifics of these customizations. All of the following work just fine in Lightning Experience, simply moving to a new location in the user interface.

- Creating custom tabs and apps with Visualforce pages.
- Creating custom buttons and links that lead to Visualforce pages.
- Creating custom actions that open with a Visualforce page.
- Overriding standard actions with Visualforce pages (with one exception that we'll get to later).
- Creating flows that use Visualforce pages.
- Packaging Visualforce pages and components.

The changes in the user interface vary from minor to significant. Features customized with Visualforce move automatically when users change between Lightning Experience and Salesforce Classic. You might need to give your users an initial orientation, but after that, they'll be happy as clams.

There are other features, such as Visualforce email templates, that use Visualforce code behind the scenes. These aren't surfaced in the user interface directly, and so they remain unchanged.

For a screenshot-heavy visual tour of where various features have moved to, see the Use Visualforce in Lightning Experience unit in the Visualforce & Lightning Experience module.

What Works, but Needs Updating and Testing

The environment in which a Visualforce page runs when Lightning Experience is enabled is different than the standard Visualforce request. The technical details get pretty complex, but the simple version is that in Lightning Experience, Visualforce pages are embedded in an HTML iframe that's displayed inside the Lightning Experience app.

This change has a number of consequences that mostly have to do with JavaScript and accessing external apps. You'll want to review your code and verify a few things before you certify your Visualforce pages for use in Lightning Experience. We're saving the nitty gritty details for the Visualforce & Lightning Experience module. For now, let's just check in at a high level so you can start to scope your review.

For starters, if you have pages or apps that use JavaScript, you'll want to review the behavior of the code. In particular, your code can't directly access the `window` global object. You can still get at it with a minor code change if you *really* need to, but there are probably better ways to accomplish those tasks using Lightning Experience app APIs instead. In particular, code which sets `window.location` directly should definitely be revised to integrate with the Lightning Experience navigation stack.

Similarly, code that assumes that it has access to the entire environment is in for a rude surprise. It still has access to the *Visualforce* part of the document, but *not* the full Lightning Experience app. That will be fine for many apps, but for those that want to be totally in charge, there's going to be some work for you to do.

If your pages make use of iframes themselves, either with `<apex:iframe>` or static HTML, being embedded into another iframe could cause some issues. In many cases "turtles all the way down" is fine. Just make sure you do extra testing here.

If your pages embed a Canvas app, and especially if you've used the Canvas APIs to integrate the app into Salesforce, allocate time for thorough testing as well. Canvas apps use an iframe, and while correctly behaving code should just work, we all know how common perfect code is in the real world.

Pages that use Remote Objects and JavaScript remoting work without requiring updates to authentication code. However, if your pages use other Salesforce APIs you might need to adapt your authentication code to make the right cross-domain request, or otherwise adjust to the new environment.

All of the above sounds both vague and hard to do but, in truth, the amount of code you're likely to need to change is small. And again, the details for developers are available in the Visualforce & Lightning Experience module.

What Doesn't Work

And so we come to the, shall we say, less pleasant part of our conversation. Fortunately, the list of what doesn't work in Visualforce for Lightning Experience is short, and we can get through it quickly.

Perhaps the most significant change, in terms of things that might be hard to work around, Visualforce overrides of standard actions are slightly different in Lightning Experience compared to Salesforce Classic. Any override for the object list action won't be accessible in Lightning Experience.

Specifically, there are six standard actions you can override for most standard and all custom objects in Salesforce Classic:

- Object tab
- Object list
- Record view
- Record edit
- Record create
- Record delete

In Lightning Experience the first two actions have been combined into one page, object home. Object home is similar to object list, along with some elements of object tab, like recent items. Other elements, such as reports or tools, have moved to other parts of the user interface.

Regardless of the user interface settings in your organization, both object tab and object list are available to be overridden in Setup. Overriding the object tab action overrides the object home page in Lightning Experience, as expected.

However, when in Lightning Experience, the object list action isn't accessible in the user interface, so there's no way to fire it. If your organization has overridden the object list action for any object, that functionality won't be available when users are using Lightning Experience. If there are essential features in that override, you'll need to find another way to make them available.

On a bit smaller scale, the `showHeader` and `sidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience. The standard Salesforce Classic header and sidebar are always suppressed, and there isn't a way to suppress the Lightning Experience header and sidebar.

A number of related lists available in Salesforce Classic aren't supported in Lightning Experience. The `<apex:relatedList>` component isn't a way around this limitation. Good try, though!

And, coming down to the really minor issues, rendering Visualforce pages as PDFs works exactly as in Salesforce Classic, without any of the Lightning Experience visual design. This is probably what you want anyway, but if you wanted to render pages into PDFs that include the Lightning Experience design, that's not possible today.

That Look-and-Feel Thing

The first thing you notice about Lightning Experience is the gorgeous, all new visual design. And if you've been developing Visualforce pages for a while, your next thought might be, how will my Visualforce pages look in Lightning Experience. The short answer is. . .well, let's sit down for this part, OK?

The short answer is, with the exception of suppressing the Salesforce Classic header and sidebar, and of being framed by the Lightning Experience user interface, Visualforce pages display unchanged in Lightning Experience.

Specifically, the HTML that's rendered by the built-in Visualforce components doesn't change when the page displays in Lightning Experience, and the Salesforce Classic stylesheets those components use is, by default, loaded by the page. The effect is that pages that use `<apex:inputField>`, `<apex:outputField>`, the `<apex:pageBlock>` components, and other coarse- and fine-grained components that match the Salesforce Classic visual design, still match that visual design. You get a little slice of Salesforce Classic in the middle of your Lightning Experience.

If, however, you've used Visualforce components that are relatively unstyled, or your own components and markup, and developed your own stylesheets instead of using the default Salesforce styles, your pages also appear unchanged, retaining the styling you worked so hard to develop.

In other words, in Visualforce for Lightning Experience we've favored stability in the visual design of existing pages, instead of trying to adapt them dynamically to Lightning Experience.

That said, if you're as excited about the new visual design as we are, there are ways for you to adopt that styling to a lesser or greater degree, depending on how much work you want to put in. We won't cover it here, but there's a whole unit, Understanding Important Visual Design Considerations, that shows the range of possibilities and the techniques to use to achieve them.

CHAPTER 28 Lightning Components in Lightning Experience

In this chapter ...

By now you've read the word "Lightning" so many times it's probably lost all meaning. Worse, we've been talking so much about both "Lightning Experience" and "Lightning components" that maybe the two terms are blurring together. Let's clear up the relationship between the two.

Remember all that information about developing following either a page-centric or app-centric model? Salesforce Classic uses a page-centric model, but Lightning Experience uses an app-centric model. It's made up of—you guessed it—components.

You can probably see where this is going. Lightning components were designed with Lightning Experience in mind. As the core Salesforce app shifts to the app-centric framework, we want you to shift along with us. We want you to think about developing on the platform in a whole new way.

You might have developed some Lightning components in Salesforce Classic. You can still use the old interface with Lightning components and all your existing component functionality transfers seamlessly into Lightning Experience.

If you haven't worked with Lightning components yet, don't worry. There's a Trailhead module, a quickstart guide, and a full developer's guide so that you can start developing in no time. Before you dive in to the technical details, let's take a second to review some basic advantages of Lightning components:

Out-of-the-Box Component Set
: Salesforce provides a number of components to bootstrap your app development.

Performance
: The component framework leverages a stateful client (using JavaScript) and a stateless server (using Apex). This structure allows the client to call the server only when absolutely necessary. With fewer calls to the server, your apps are more responsive and efficient.

Event-Driven Architecture
Events are key to the Lightning component framework. Components listen to application and component events and respond accordingly.

Rapid Development
The simple markup and pre-made components mean that you can get applications out the door faster than ever. Particularly if you're comfortable with Visualforce markup, learning component markup is a breeze.

Device-Aware and Cross-Browser Compatibility
A huge advantage of Lightning components is that you don't have to worry about compatibility across devices and browsers. The component framework handles this work for you.

Considerations for Use

We've covered some of the considerations for using Lightning components. You probably don't want to switch to Lightning components with in-progress Visualforce projects. You also want to stick with Visualforce if you need to do things like render PDFs from a page. Visualforce hasn't gone away, and remains a foundational part of developing on the Salesforce platform.

The Lightning Component framework is, relatively speaking, the new kid on the block—but this new kid has skills. As of the Spring '19 release (API version 45.0), you can build Lightning components using two programming models: the Lightning Web Components model and the original Aura Components model. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page. While Lightning components have a few specific limitations, for the most part, they're ready to go. There are many situations where you should consider making the switch to developing with Lightning components. Salesforce mobile development, for example, is a great place to use Lightning components. Also use Lightning components for new projects and any project involving highly interactive applications.

Let's make a list! Where exactly can you use Lightning components? You have many options.

Lightning Experience

We said it earlier, but don't want you to get the wrong idea by leaving it off this list. Lightning Experience and Lightning components are two great tastes that taste great together.

Salesforce App

We're repeating this one often because it's important: use Lightning components for your mobile development. When you're using a mobile device, you don't want to make a call to the server every time a user presses a button. Using Lightning components vastly improves mobile app performance.

Standalone Apps

If you used Lightning components in Salesforce Classic, you probably made at least one standalone Lightning app. Lightning App Builder lets you declaratively create apps with standard components ranging from buttons to Canvas apps. Alternatively, use the Developer Console to create apps made up of both standard and custom Aura components. See the *Lightning Aura Components Developer Guide* for more information.

Visualforce Pages

This capability is perfect for Salesforce developers who are Visualforce veterans. If you're not quite ready to commit to a full Lightning app, smooth the transition by integrating Aura components into Visualforce pages. This task only requires a few lines of markup and gives you a huge amount of flexibility. See the *Lightning Aura Components Developer Guide* for more information about Aura components for Visualforce.

Anywhere!
Lightning Out, currently available in beta, lets you run your Aura components and apps, well, pretty much anywhere you can serve a web page. Whether it's a Node.js app running on Heroku, a department server inside the firewall, or even SharePoint (yes, SharePoint), build your custom app with Lightning components and run it wherever your users are.

Screen Flows
If you haven't taken advantage of screen flows yet, you're missing out on a powerful way to guide your users through a business process. Customize the look-and-feel and functionality of your flow's screens by adding Aura components to them.

As much engineering effort as we've put into making Lightning components a framework you can use to create applications for the next decade, we're not done. There's still a few places where you can use Visualforce to customize Salesforce but you can't yet use Lightning components. Stay tuned to this channel.

CHAPTER 29 Salesforce Partners, Packaging, and AppExchange

In this chapter ...

If you're a Salesforce AppExchange partner (ISV), you're probably wondering how Lightning Experience affects your development and release processes. Do your existing products still work? Do you have to make major changes to existing procedures? What about future releases?

These are valid concerns, but we have good news. Many of the tools that you use to build, test, and distribute your products are available in Lightning Experience. We also have resources to guide you as you update your products to be Lightning Ready.

Partner Tools in Lightning Experience

First, let's talk tools. This conversation is bittersweet. Some of the tools that you use are available in Lightning Experience, while others aren't supported there yet. Here's an overview of what's supported right now.

Feature	**Supported In**
• Environment Hub • License Management App (LMA) • Checkout Management App (CMA) • Channel Order App (COA)	Both Salesforce Classic and Lightning Experience.
• Trialforce • Usage Metrics Visualization App	Salesforce Classic only.

This means you can go to Lightning Experience if you want to:

- Create orgs for development, testing, and demos
- Manage all your orgs from a single location
- License and support apps published on the AppExchange
- View and report on subscription data for products sold using AppExchange Checkout
- Submit and manage orders with the Channel Order App

As Lightning Experience continues to mature, more tools will become available.

Packaging Apps in Lightning Experience

When you're ready to distribute an app, Lightning component, or other product, Lightning Experience is up to the task. The package manager lets you create and manage packages from Setup, just like you could in Salesforce Classic. Whether you're creating managed packages in a Developer Edition or unmanaged packages in an Enterprise Edition, the new packaging experience feels right at home in your development process.

AppExchange and Lightning Experience

If you offer a product on AppExchange, you probably have questions about getting Lightning Ready. We consider products Lightning Ready when all end-user use cases work as expected in Lightning Experience.

So, how long do you have to get Lightning Ready? The answer depends on the type of product and your distribution agreement with Salesforce. New products sold using AppExchange Checkout and new free products must be Lightning Ready before their security review. Other agreement types may require Lightning Experience readiness depending on when you signed the agreement. For existing products, we encourage you to get Lightning Ready as soon as possible. For more information and resources, go to the Lightning Ready page on the Salesforce Partner Community.

If you're an AppExchange customer, look for the Lightning Ready certification on a product's listing to verify that it's compatible with Lightning Experience. You can use products that aren't certified as Lightning Ready in Lightning Experience, but there's no guarantee that they'll work as expected. It's best to use them in Salesforce Classic.

CHAPTER 30 Understand Changes to Other Development Tools

In this chapter ...

This last unit is a bit of a grab bag. We've already covered all of the "hard" stuff, so at this point you're in the home stretch. Can you feel it? That's the itch you get when you're close to a new badge. Let's do this.

Installed Packages in Lightning Experience

Managing your installed packages in Lightning Experience is the same as it ever was. The Installed Packages landing page is available in the Lightning Experience setup area. It looks and works the same way it does in Salesforce Classic.

Of course, finding and using this page isn't the only thing on your mind. You're also wondering whether your packages installed from AppExchange still work in Lightning Experience.

The best answer we can give is **maybe**. Apps listed on AppExchange are marked with a "Lightning Ready" sash if they're fully compatible with Lightning Experience. Check the listing to see if an app is Lightning Ready. If it's not, you can still try to use it in Lightning Experience, but we recommend sticking to Salesforce Classic to prevent unexpected behavior.

The API and Apex in Lightning Experience

As a developer, one of your most important tools on any platform is the API. As a Salesforce developer, Apex is just as important to your success.

We've kept our promise that we won't do anything that breaks your dependencies on our APIs. Your Apex code and queries continue to function as expected, regardless of whether you're using Lightning Experience or Salesforce Classic. It really is just that simple. Breathe a sigh of relief.

Authentication and Security in Lightning Experience

Regardless of the user experience you're developing for, security is still Salesforce's top priority. Lightning Experience doesn't fall short of our promise to keep your organization's data safe.

Continue to treat authentication and security as you do when developing for Salesforce Classic. The only difference in access controls between Salesforce Classic and Lightning Experience is in the App Launcher. The App Launcher is available by default to all users in your organization. While this change is of concern mostly to administrators, it's important to work with your Salesforce admin to ensure that your development work is only seen by the people who are supposed to see it.

Canvas for Lightning Experience

Canvas allows you to easily integrate third party applications in Salesforce. Canvas functionality in Lightning Experience is the same as in Salesforce Classic. You can still embed Canvas apps in Visualforce pages, the

Salesforce mobile app, and everywhere else they're supported—with the added bonus that you can integrate Canvas apps in Lightning Components!

Salesforce Mobile for Lightning Experience

Lightning Experience and Salesforce mobile are like peanut butter and jelly. They're made for each other. Your mobile development practices in Lightning Experience are the same as they were in Salesforce Classic.

When we say the two were made for each other, we mean it. You might be familiar with the `sforce.one` JavaScript object. In the past, it was used as an event mechanism for navigation in Salesforce mobile development. Now, you can also use it for navigation in Lightning Experience. See the Salesforce Mobile App Developer's Guide for more information.

Mobile SDK for Lightning Experience

By now you're probably tired of reading overview information and are ready to dive in to the nitty-gritty technical details. We won't keep you much longer. Literally just another two sentences!

As its own client, or front end, to Salesforce the Mobile SDK isn't impacted by Lightning Experience. If you use Mobile SDK to develop mobile apps, you can rest easy.

MORE ABOUT REPLACING JAVASCRIPT BUTTON FUNCTIONALITY FOR LIGHTNING EXPERIENCE

CHAPTER 31 JavaScript Buttons: It's Time to Move On

In this chapter ...

We know that you love JavaScript buttons and have been using them in Salesforce Classic for years. In fact, you might be reluctant to migrate to Lightning Experience because JavaScript buttons aren't supported. But Lightning Experience offers so much more than Salesforce Classic, and it is the future. We realize that you expect and rely on us to always migrate existing functionality to new features and UI, but in the case of JavaScript buttons, we believe that the future is brighter in Lightning Experience, even without JavaScript button support. And in this module, we show you why.

JavaScript buttons and links are types of actions in the Salesforce Classic UI that let you create inline JavaScript code that can be invoked via a button or link embedded on a record or list page. For example, maybe you prepopulate new records with data upon creation and update values in fields based on other logic. Or maybe you're a Salesforce partner who uses custom buttons to integrate with your platform.

If JavaScript buttons are so useful, why don't we support them in Lightning Experience? Because there are significant security challenges with combining untrusted JavaScript from multiple sources and authors together with the application source code, while maintaining trust.

We'll cover those security and functional challenges, and share with you the alternatives to JavaScript buttons that are mobile- and Lightning-friendly. We'll also look at features in Salesforce that you can use to migrate the functionality that you've built using custom buttons.

We're committed to solving the problem of client-side customization and integration. Let us show you a new approach to thinking about JavaScript button functionality in Lightning Experience.

JavaScript Button Security Issues and Use Cases

One of the coolest benefits of Lightning Experience is that you can add your custom Lightning components to record, Home, and other pages. For example, you might choose to add a map component to your Account record pages. Or, you could provide a component for your AppExchange app that can be added to the Home page or an Opportunity record.

However, without some safeguards, the components have access to each other's data, shared access to the window and event structures, and access to any client-side API. A partner's component for HIPAA compliance or financial information, for example, could be accessed by a component from a different source when both components are on the same page. As you can imagine, this cross-component access could lead to security and regulatory issues.

What's Up with Inline JavaScript

Before we discuss the safeguards that Salesforce has in place for Lightning component security, let's highlight some of the issues with in-line JavaScript. JavaScript is a loosely typed programming language, supported by all modern web browsers without a plug-in. It can persist data and state through cookies and storage APIs, and it can access events, URLs, and cookies through the browser. What makes JavaScript both useful and dangerous is that it has full access to the Document Object Model (DOM) and Browser Object Model (BOM).

With access to the DOM, a programmer can add, change, or delete almost anything found in an HTML or XML document. In the right hands, this is useful because JavaScript provides an API for working with text, dates, and regular expressions, so it's easy to add client-side functionality with JavaScript snippets that enhance the base user interface. However, this is also a significant vulnerability because with Cross Site Scripting (XSS), malicious actors can gain access via JavaScript to the DOM or BOM and wreak havoc.

When a website enables dynamic content, hackers can use XSS to inject malicious client-side code into the web pages that are viewed by normal users. The hackers can then harness a user's session and cookies to run scripts to extract data, log keystrokes, manipulate form entries, and even access APIs.

Lightning Locker: Making Lightning Components More Secure

The good news is that Salesforce is already working on a solution to make Lightning components more secure and restrict JavaScript's unfettered access. This solution is Lightning Locker, which uses various technologies and techniques that are intended to prevent:

- XSS and similar security issues

- Unrestricted DOM access
- Calls to undocumented/private APIs

At the same time, Lightning Locker's features also enable:

- Client-side API versioning
- Faster security review (AppExchange)
- Better JavaScript development practices
- Easy updates to security features and policies

So you now know that Lightning components are built to be more secure. But how can you benefit from using them, and how can you re-create your JavaScript button functionality within Lightning Experience? We'll show you shortly. But first, let's look at how you might be using JavaScript buttons in Salesforce Classic.

What People Are Doing with JavaScript Buttons

We heard from many customers, some with hundreds of JavaScript buttons in their orgs. We also talked to partners about their JavaScript button use cases. We collated what we learned into a broader set of operations. Here are the most common use cases for JavaScript buttons.

- Use or manipulate values on a record *before* the save
  - Validate fields—ensure that values are populated and/or meet criteria
  - Prefill values based on inputs in other fields
  - Redirect to a Visualforce page based on input values
  - Display confirmation pop-up screens
- Create records with prepopulated values
- Trigger flows built in Flow Builder
- Call out to Salesforce or external APIs
- Integrate with third parties
- Perform mass actions on records in a list
- Direct methods and procedures via feedback pop-up screens for users

There are more scenarios, and some use cases that are so specific to an org that they're impossible to categorize. Coming up, we'll cover features that you can use to address all the use cases we mentioned, and migrate JavaScript button functionality to the Salesforce mobile app and Lightning Experience.

CHAPTER 32 You Have Buttons? We Have Alternatives

In this chapter ...

Previously, we explained why it's time to migrate to Lightning Experience, where great new features let you move past JavaScript buttons. Next, we'll explain how you can easily migrate your custom JavaScript button functionality using solutions that work in both Salesforce Classic and Lightning Experience.

This table maps use cases for JavaScript buttons to alternate—and in most cases, better—solutions in Lightning.

JavaScript Button Top Use Cases	Lightning Alternatives	Declarative/Programmatic
Validate fields (presave)	Quick actions (using default values and/or formulas)	D
	Apex triggers	P
Create records with prepopulated values	Quick actions (using default values and/or formulas)	D
Redirect to a record page	Custom URL buttons	D
Redirect to a Visualforce page	Visualforce quick actions	P
	Lightning actions	P
Prefill values based on inputs	Lightning actions	P
Confirmation pop-up screens	Lightning actions	P

JavaScript Button Top Use Cases	Lightning Alternatives	Declarative/Programmatic
API calls (Salesforce and third-party)	Lightning actions	P
Feedback pop-up screens	Lightning actions	P
Third-party integration	Lightning actions	P
Mass actions on list view records	Custom Visualforce buttons on list views	P

As you can see, Salesforce offers several declarative tools for converting the functionality of your JavaScript custom buttons.

Quick Actions

Quick actions support many of the common uses of JavaScript buttons. Quick actions can be based on a specific object or can be global, meaning that they're more generic and accessible from any record or the Chatter feed. There's a quick action to do just about anything in Salesforce. Here are some examples.

Validate Field Values

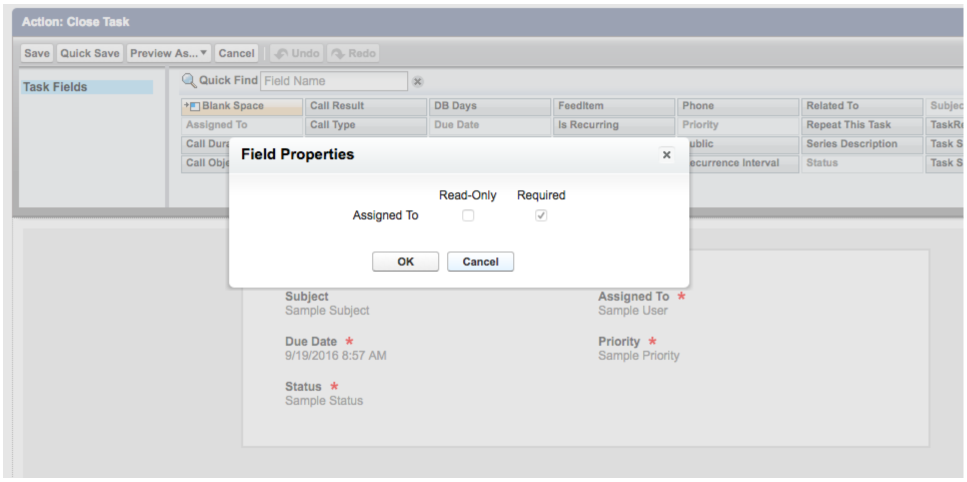
Sometimes you want to make sure that certain fields are filled in or populated with specific criteria when your users create or update records.

Let's say you want to create an action for closing a task without requiring users to go to a full edit page. But you also want to make sure that the task has a due date before it can be closed.

You can fulfill all these requirements by creating a quick action for the Task object.

1. From the Object Manager in Setup, click **Task**, then click **Buttons, Links, and Actions**.
2. Click **New Action**.
3. For Action Type, select **Update a Record**.
4. For Label, type in `Close Task`.
5. Click **Save**.

Now we select the fields that we want to appear on the Close Task quick action. You can easily make a field required or read-only through its field properties.



	Read-Only	Required
Assigned To	☐	✓

Subject
Sample Subject

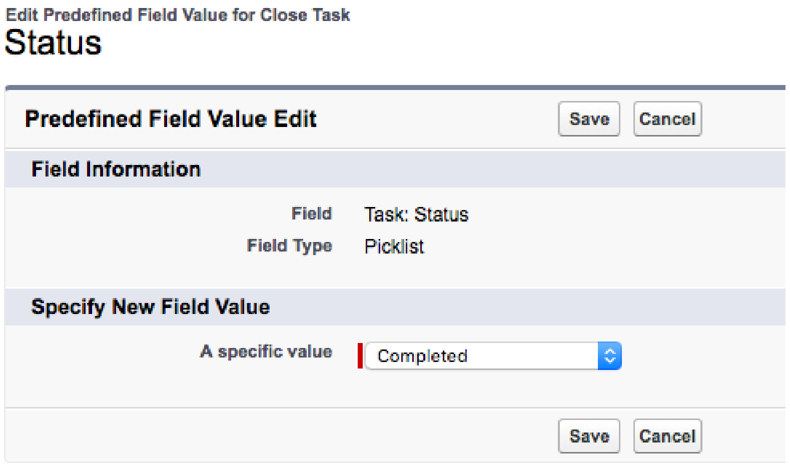
Assigned To *
Sample User

Due Date *
9/19/2016 8:57 AM

Priority *
Sample Priority

Status *
Sample Status

After you set the fields on the action layout, you can add predefined field values for any of the fields on the task record. In this example, we marked the Due Date field as required. We also added a predefined value for the Status field to be changed to Completed.


Edit Predefined Field Value for Close Task

Status

Predefined Field Value Edit

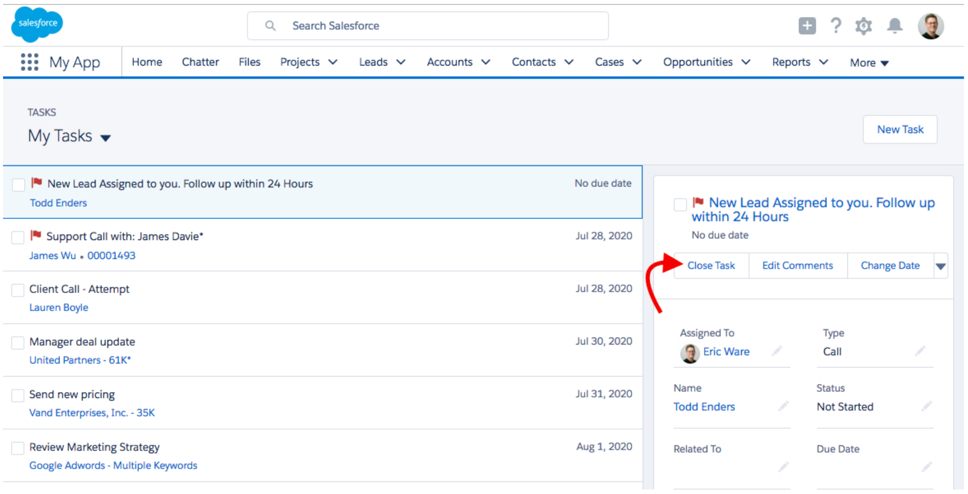
Field Information

Field	Task: Status
Field Type	Picklist

Specify New Field Value

A specific value: Completed

Now that we're done configuring the action, we add it to the Task page layout. Then users can access it from a task record page in both Lightning Experience and the Salesforce mobile app. Here's an example of a Close Task action on the Tasks page.



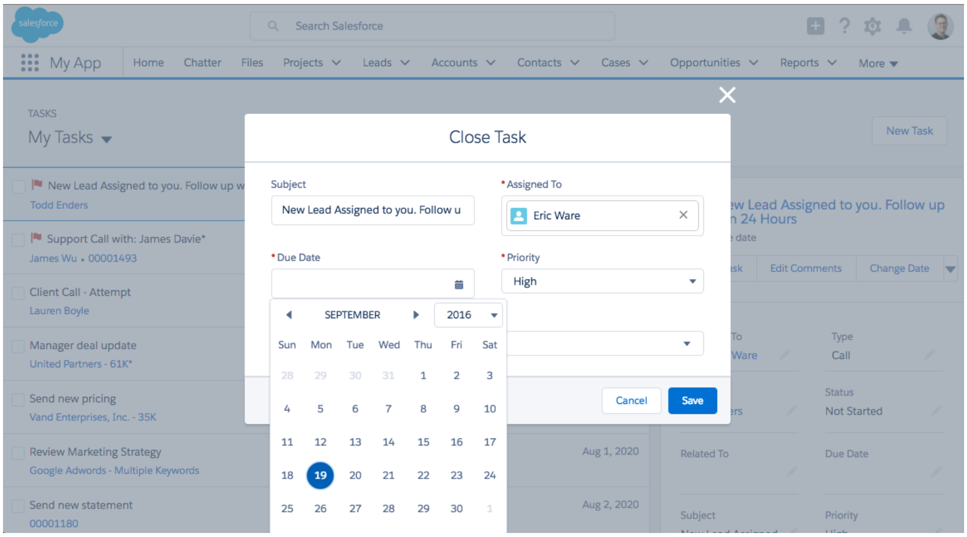
New Lead Assigned to you. Follow up within 24 Hours Todd Enders	No due date
Support Call with: James Davie* James Wu • 00001493	Jul 28, 2020
Client Call - Attempt Lauren Boyle	Jul 28, 2020
Manager deal update United Partners - 61K*	Jul 30, 2020
Send new pricing Vand Enterprises, Inc. - 35K	Jul 31, 2020
Review Marketing Strategy Google Adwords - Multiple Keywords	Aug 1, 2020

New Lead Assigned to you. Follow up within 24 Hours

No due date

Assigned To: Eric Ware	Type: Call
Name: Todd Enders	Status: Not Started
Related To	Due Date

Clicking **Close Task** brings up the action, which the user can quickly act upon and save.



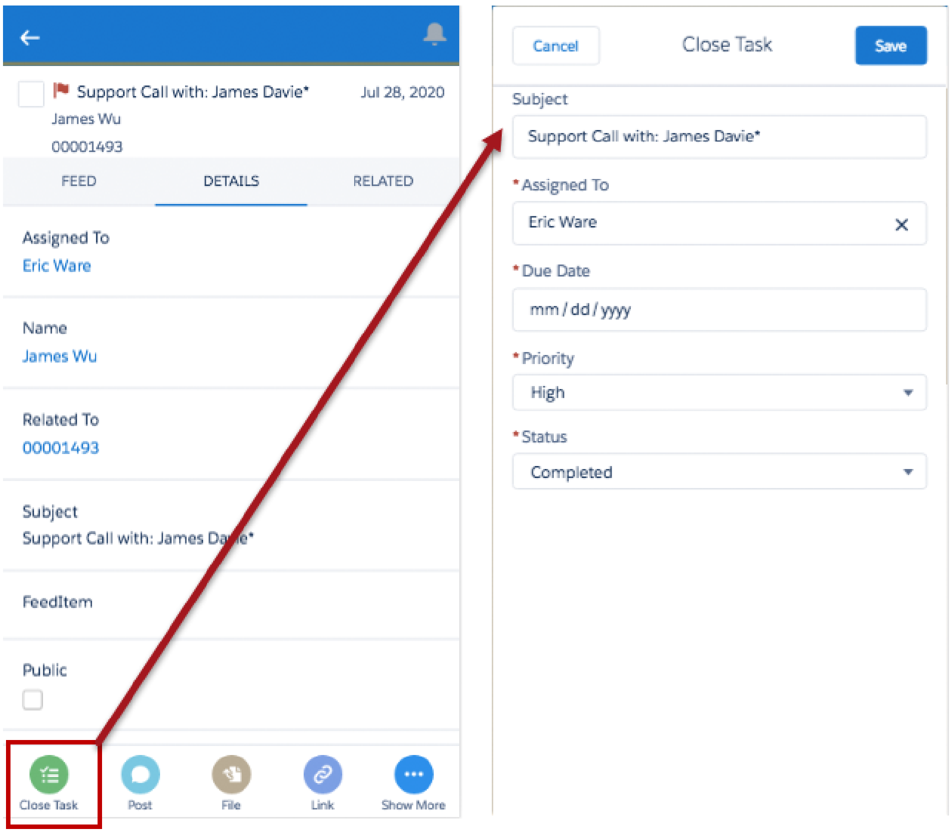
New Lead Assigned to you. Follow up w
Todd Enders
Support Call with: James Davie*
James Wu • 00001493
Client Call - Attempt
Lauren Boyle
Manager deal update
United Partners - 61K*
Send new pricing
Vand Enterprises, Inc. - 35K
Review Marketing Strategy
Google Adwords - Multiple Keywords
Aug 1, 2020
Send new statement
00001180
Aug 2, 2020

Subject: New Lead Assigned to you. Follow u
*Assigned To: Eric Ware
*Due Date:
*Priority: High

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Type: Call
Status: Not Started
Related To
Due Date
Subject
Priority

Here’s what it looks like in the Salesforce mobile app.


Support Call with: James Davie*
Jul 28, 2020
James Wu
00001493

Assigned To
Eric Ware

Name
James Wu

Related To
00001493

Subject
Support Call with: James Davie*

FeedItem

Public

Subject
Support Call with: James Davie*

*Assigned To
Eric Ware

*Due Date
mm / dd / yyyy

*Priority
High

*Status
Completed

Prepopulate Fields with Values

A more advanced use case is when you want to let users create a record, but you also want one or more of the fields to be populated automatically based on values in a dependent field.

For example, let's say your inside sales team's typical quarterly quota is one fourth of their customers' revenues from last year, increased by 10%. Because that simple formula doesn't always apply, you want to prefill the opportunity amount with the value, but also let the user modify it. Create an action so your users can modify the field quickly and efficiently without going to the full opportunity record page.

To create this sample action, you follow the same steps as before for creating the opportunity quick action.

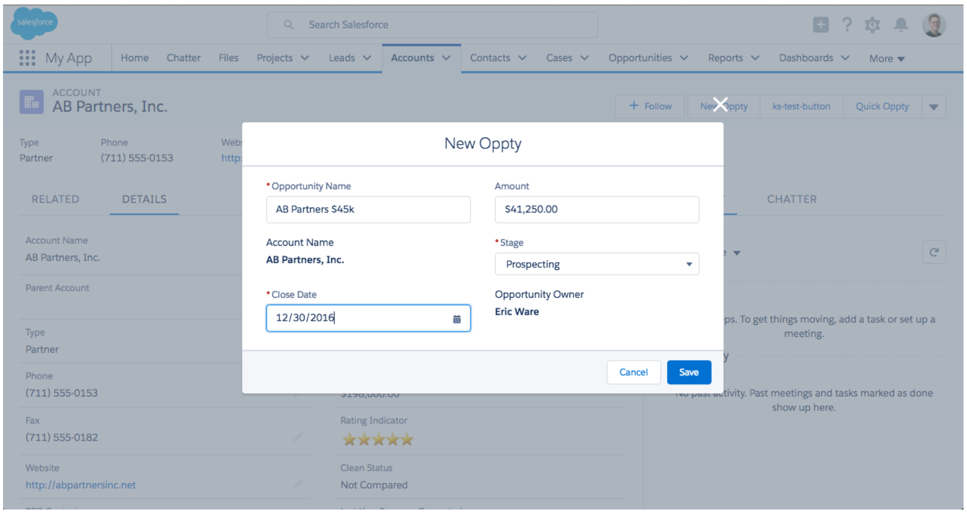
1. From the Object Manager in Setup, click **Account**, then **Buttons, Links, and Actions**.
2. Click **New Action**.
3. For Action Type, select **Create a Record**.

4. For Target Object, select **Opportunity**.
5. Pick the appropriate Record Type.
6. For Label, enter `New Oppty`.
7. Click **Save**.

Once you've picked the fields for the action layout, you can add predefined values for the Amount field. In our example, we've used this formula:

```
Account.Last_Year_Revenue_Generated__c  * 1.10  / 4
```

Add this action to the Account page layout. When users invoke it, they see a value prepopulated in the field that they can accept or override.



Type: Partner
Phone: (711) 555-0153

New Oppty

*Opportunity Name: AB Partners $45k
Amount: $41,250.00
Account Name: AB Partners, Inc.
*Stage: Prospecting
*Close Date: 12/30/2016
Opportunity Owner: Eric Ware

Account Name: AB Partners, Inc.
Parent Account
Type: Partner
Phone: (711) 555-0153
Fax: (711) 555-0182
Website: http://abpartnersinc.net
Rating Indicator: ★★★★★
Clean Status: Not Compared

ps. To get things moving, add a task or set up a meeting.

No past activity. Past meetings and tasks marked as done show up here.

The cool thing about this action is that you can pull data from the opportunity's account to use when creating the opportunity. Salesforce also supports record traversal, so if you have hierarchical accounts, you can pull the revenues from the parent account like this:

```
Account.Parent.Last_Year_Revenue_Generated__c
```

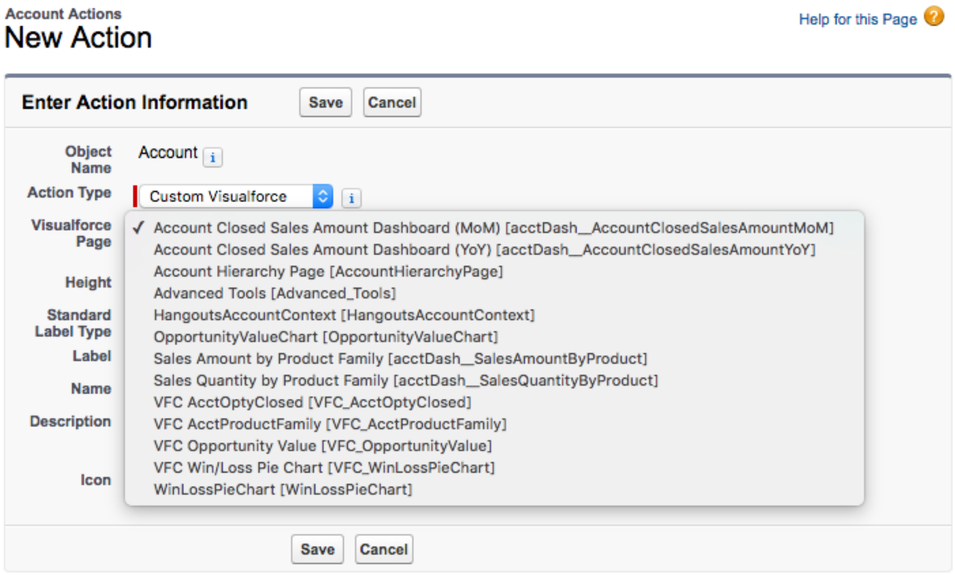
Don't worry about remembering the formula formats; quick actions are declarative and use the Salesforce formula builder.

Redirect to a Visualforce Page Based on Input Values

You can create Visualforce pages to enhance your business processes. Users can navigate to these Visualforce pages in various ways, such as with custom buttons, action overrides, and tabs.

One benefit of Visualforce pages is that by using the standard controller, you can create customized record pages and add prevalidation, prepopulated fields, formulas, and much more.

JavaScript buttons are commonly used in Salesforce Classic to read and pass values from a record into a URL that then redirects users to a Visualforce page. You can also give your users access to Visualforce pages via quick actions. Creating Visualforce quick actions is easy, and a similar process to what we've already covered. The only difference is that you select Custom Visualforce as the action type.


Account Actions
New Action
Help for this Page

Enter Action Information [Save] [Cancel]

Object Name: Account
Action Type: Custom Visualforce
Visualforce Page:
- ✓ Account Closed Sales Amount Dashboard (MoM) [acctDash__AccountClosedSalesAmountMoM]
- Account Closed Sales Amount Dashboard (YoY) [acctDash__AccountClosedSalesAmountYoY]
- Account Hierarchy Page [AccountHierarchyPage]
- Advanced Tools [Advanced_Tools]
- HangoutsAccountContext [HangoutsAccountContext]
- OpportunityValueChart [OpportunityValueChart]
- Sales Amount by Product Family [acctDash__SalesAmountByProduct]
- Sales Quantity by Product Family [acctDash__SalesQuantityByProduct]
- VFC AcctOptyClosed [VFC_AcctOptyClosed]
- VFC AcctProductFamily [VFC_AcctProductFamily]
- VFC Opportunity Value [VFC_OpportunityValue]
- VFC Win/Loss Pie Chart [VFC_WinLossPieChart]
- WinLossPieChart [WinLossPieChart]

Height
Standard Label Type
Label
Name
Description
Icon

[Save] [Cancel]

For object-specific Visualforce quick actions, you must include the `standardController` for the object in your Visualforce page to gain access to the record data and have the Visualforce page appear in the quick action picklist.

Custom URL Buttons and Links

Maybe you use JavaScript buttons for navigation, redirecting users to another page with the command `window.open(URL)` and some variables. In most cases, you can use custom URL buttons or links in Lightning Experience instead.

Here's a list of different URL buttons and links, and their redirect behavior in Lightning Experience.

Custom URL Button or Link	Lightning Experience Behavior
External URL	URL opens in new tab

Custom URL Button or Link	Lightning Experience Behavior
`www.google.com`	
Relative Salesforce URL, View `/{!Account.Id}`	Record home page opens in existing tab
Relative Salesforce URL, Edit `/{!Account.Id}/e`	Edit overlay pops up on the existing page
Relative Salesforce URL, List `/001/o`	Object home page opens in existing tab
$Action URL, View `{!URLFOR($Action.Account.View, Account.Id)}`	Record home page opens in existing tab
$Action URL, Edit `{!URLFOR($Action.Account.Edit, Account.Id)}`	Edit overlay pops up on the existing page

Apex Triggers

You might be familiar with Apex triggers; they've been supported on our platform for years. Apex triggers can be configured to execute before or after a user clicks Save on a record.

When you need presave validation, calculation, and population of fields, consider using Apex triggers. They're especially useful for third-party integration, because the rules are enforced through the Salesforce API across Salesforce Classic, Lightning Experience, and the Salesforce mobile app.

For more information on Apex triggers, you can check out the *Apex Developer Guide* or earn the Apex Triggers badge.

Custom Visualforce Buttons

Another great feature in Lightning Experience is support for using Visualforce buttons on list views. With this feature, you can use existing Visualforce actions in Lightning and work with multiple records in lists. Here's how.

1. Create your Visualforce page.

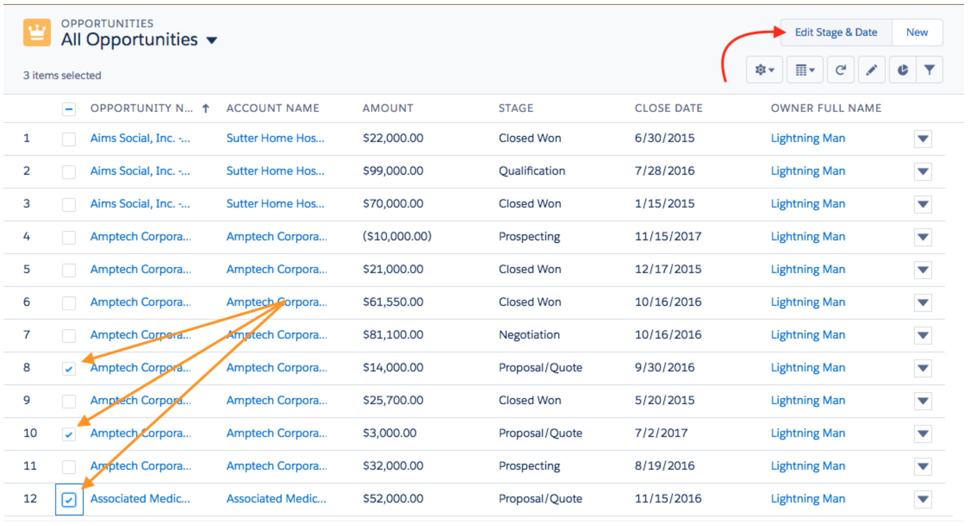
   Here is sample code for editing the Stage and Close dates for multiple opportunities:

```
<apex:page standardController="Opportunity"
recordSetVar="opportunities" extensions="tenPageSizeExt">
   <apex:form>
      <apex:pageBlock title="Edit Stage and Close Date"
mode="edit">
         <apex:pageMessages />
         <apex:pageBlockButtons location="top">
            <apex:commandButton value="Save" action="{!save}"/>
            <apex:commandButton value="Cancel"
action="{!cancel}"/>
         </apex:pageBlockButtons>
         <apex:pageBlockTable value="{!selected}" var="opp">
            <apex:column value="{!opp.name}"/>
            <apex:column headerValue="Stage">
               <apex:inputField value="{!opp.stageName}"/>
            </apex:column>
            <apex:column headerValue="Close Date">
               <apex:inputField value="{!opp.closeDate}"/>
            </apex:column>
         </apex:pageBlockTable>
      </apex:pageBlock>
   </apex:form>
</apex:page>
```

2. Create a custom button that references your Visualforce page.
3. Add the action to your list view.

   Note: Mass actions aren't supported on the Recently Viewed records list. They are only available on list views.

Example: Here's how that action shows up in Lightning Experience.



		OPPORTUNITY N...	ACCOUNT NAME	AMOUNT	STAGE	CLOSE DATE	OWNER FULL NAME
1		Aims Social, Inc. -...	Sutter Home Hos...	$22,000.00	Closed Won	6/30/2015	Lightning Man
2		Aims Social, Inc. -...	Sutter Home Hos...	$99,000.00	Qualification	7/28/2016	Lightning Man
3		Aims Social, Inc. -...	Sutter Home Hos...	$70,000.00	Closed Won	1/15/2015	Lightning Man
4		Amptech Corpora...	Amptech Corpora...	($10,000.00)	Prospecting	11/15/2017	Lightning Man
5		Amptech Corpora...	Amptech Corpora...	$21,000.00	Closed Won	12/17/2015	Lightning Man
6		Amptech Corpora...	Amptech Corpora...	$61,550.00	Closed Won	10/16/2016	Lightning Man
7		Amptech Corpora...	Amptech Corpora...	$81,100.00	Negotiation	10/16/2016	Lightning Man
8	✓	Amptech Corpora...	Amptech Corpora...	$14,000.00	Proposal/Quote	9/30/2016	Lightning Man
9		Amptech Corpora...	Amptech Corpora...	$25,700.00	Closed Won	5/20/2015	Lightning Man
10	✓	Amptech Corpora...	Amptech Corpora...	$3,000.00	Proposal/Quote	7/2/2017	Lightning Man
11		Amptech Corpora...	Amptech Corpora...	$32,000.00	Prospecting	8/19/2016	Lightning Man
12	✓	Associated Medic...	Associated Medic...	$52,000.00	Proposal/Quote	11/15/2016	Lightning Man

As you can see from our examples, Salesforce has a great set of features that allow you to migrate your JavaScript button functionality and move to Lightning Experience.

Now, you may have a lot of JavaScript buttons that have accumulated in your org over the years. You might expect that the migration or conversion process will take a long time. But the job might not be as hard as you think. We did an analysis of JavaScript buttons in the internal Salesforce org that all our employees use. We found that many buttons were obsolete or rarely invoked by users. Others were simply duplicates—the same button but on different objects. After going through the list, we discovered that many of the JavaScript buttons could be converted to the solutions we've looked at so far. For the remainder, we could address quite a few with the new Lightning Actions.

You probably noticed Lightning actions in the list of programmatic solutions as a common replacement for many of the JavaScript button use cases. Lightning actions are easy to build, because they're based on the quick action framework. You set them up similar to Visualforce quick actions. We'll cover Lightning actions more in depth next.

CHAPTER 33 Lightning Actions: Smart, Fast, and Mobile

In this chapter ...

We've covered several solutions that work in both Lightning Experience and Salesforce Classic and that are great replacements for JavaScript buttons. But we recognize that those declarative and programmatic solutions don't address every use case. The good news is that there's something that addresses many more. Introducing Lightning actions: quick actions that call Lightning components.

Note: As of the Spring '19 release (API version 45.0), you can build Lightning components using two programming models: the Lightning Web Components model and the original Aura Components model. Lightning web components and Aura components can coexist and interoperate on a page. This content covers Aura components.

Lightning actions are built on the existing Lightning Component framework. You can easily convert your existing Aura components to actions and use them in the Salesforce mobile app and Lightning Experience.

You make an Aura component invokable as an action by adding one of two interfaces to the component: `force:lightningQuickAction` or `force:lightningQuickActionWithoutHeader`. The first interface adds the standard header with Save and Cancel buttons to the Lightning action overlay, and the other doesn't.

Another useful interface for a Lightning action is `force:hasRecordId`, which provides the record context to the component when it's invoked from a record page. If you set up your Lightning action as a global quick action, you don't need the record context. However, if you want to access a record's data or metadata, you must implement `force:hasRecordId`.

Note: If you haven't built Aura components before, visit the Lightning Development Center, do the Aura Components Quick Start project, and complete the Aura Components Basics module.

Let’s dive into Lightning actions. Next, we’ll talk about some JavaScript button functionality that can be addressed with Lightning actions instead.

- Populating fields based on user input and providing feedback messages during data entry
- Integrating third-party APIs

Populate Fields Based on User Input and Provide Feedback Messages to Users

Let's say you use JavaScript buttons to validate or manipulate data and give your users feedback or instructions when they work with records. Here's an example to show how easily you can validate or manipulate data in a Lightning component before your users create or update a record.

We've built a custom object in our sample org called Case Study, which we use to track different research projects. When we add new test users, we capture their names and email addresses. Since the email address is our primary contact method, we want to make sure it's entered correctly. We also want to create a unique nickname for test users so that they can remain anonymous.

We're going to create a Lightning action that displays the name and email address fields, validates the email address, and uses the first name and a random number to automatically create the nickname.

Here's the code for the component called by this Lightning action.

- `CreateUser.cmp`—The Lightning component that you see when you open the action. It contains the UI implementation, which includes the text fields, buttons, action title, and so on.
- `CreateUserController.js`—The controller that listens to the component and any UI events that take place, such as the initial load, text input, and button clicks. Its function is to delegate these events to the helper class, `CreateUserHelper.js`.
- `CreateUserHelper.js`—The code that deals with all the business logic, such as validating the password and email fields.

CreateUser.cmp

```
<aura:component
implements="force:lightningQuickActionWithoutHeader,force:hasRecordId">

    <aura:attribute name="user" type="Test_User__c" default="{}"/>
    <aura:attribute name="hasErrors" type="Boolean"
description="Indicate whether there were failures or not" />
    <aura:attribute name="caseStudy" type="String" />
    <aura:attribute name="recordId" type="String"/>
    <aura:attribute name="errorFromCreate" type="String"/>

    <!-- <aura:handler name="init" value="{!this}" action="{!c.init}"
 />  -->

    <force:recordData aura:id="frd" mode="EDIT" layoutType="FULL"/>
    <div class="slds-page-header" role="banner">
     <p class="slds-text-heading--label">Case Study</p>
        <h1 class="slds-page-header__title slds-m-right--small
slds-truncate slds-align-left" title="Case Study
Title">{!v.caseStudy}</h1>
    </div>
    <br/>

    <aura:if isTrue="{!v.hasErrors}">
        <div class="userCreateError">
          <ui:message title="Error" severity="error" closable="true">

                Please review the error messages.
            </ui:message>
        </div>
    </aura:if>

    <div class="slds-form--stacked">

        <div class="slds-form-element">
            <label class="slds-form-element__label"
for="firstName">Enter first name: </label>
            <div class="slds-form-element__control">
              <ui:inputText class="slds-input" aura:id="firstName"
value="{!v.user.First}" required="true" keydown="{!c.updateNickname}"
 updateOn="keydown"/>
            </div>
        </div>

        <div class="slds-form-element">
          <label class="slds-form-element__label" for="lastName">Enter
 last name: </label>
            <div class="slds-form-element__control">
              <ui:inputText class="slds-input" aura:id="lastName"
value="{!v.user.Last}" required="true" />
            </div>
        </div>

        <div class="slds-form-element">
          <label class="slds-form-element__label" for="nickname">Enter
 nickname: </label>
            <div class="slds-form-element__control">
              <ui:inputText class="slds-input" aura:id="nickname"
```

```
value="{!v.user.Nickname}" required="false"/>
            </div>
        </div>

        <div class="slds-form-element">
        <label class="slds-form-element__label" for="userEmail">Enter
 user's email:</label>
            <div class="slds-form-element__control">
          <ui:inputEmail class="slds-input" aura:id="userEmail"
value="{!v.user.Email__c}" required="true"/>
            </div>
        </div>

        <div class="slds-form-element">
         <label class="slds-form-element__label"
for="userPassword">Enter user's password:</label>
            <div class="slds-form-element__control">
          <ui:inputSecret class="slds-input" aura:id="userPassword"
value="{!v.user.Password__c}" required="true"/>
            </div>
        </div>

        <div class="slds-form-element">
         <ui:button class="slds-button slds-button--neutral"
press="{!c.cancel}" label="Cancel" />
         <ui:button class="slds-button slds-button--brand"
press="{!c.saveUserForm}" label="Save User" />
        </div>
    </div>

</aura:component>
```

CreateUserController.js

```
({
    /**
     * Auto generate the username from firstName on tab
     */
    updateNickname: function(component, event, helper) {
        // Update the nickname field when 'tab' is pressed
        if (event.getParams().keyCode == 9) {
         var nameInput = component.find("firstName");
         var nameValue = nameInput.get("v.value");
         var nickName = component.find("nickname");
            var today = new Date();
         nickName.set("v.value", nameValue + today.valueOf(today));

        }
    },

    /**
     * Capture the Inputs and invoke the helper.save with the input
params
     */
 saveUserForm : function(component, event, helper) {
        var name = component.get("v.user.First");
        var last = component.get("v.user.Last");
        var password = component.get("v.user.Password__c");
        var email = component.get("v.user.Email__c");
        var nickname = component.get("v.user.Nickname");

        var passwordCmp = component.find("userPassword");
        var emailCmp = component.find("userEmail");

        helper.validatePassword(component, event, helper);
        helper.validateEmail(component, event, helper);

        if (passwordCmp.get("v.errors") == null &&
emailCmp.get("v.errors") == null) {
            component.set("v.hasErrors", false);
         helper.save(component, name + " " + last, password, email,
nickname);
        } else {
            component.set("v.hasErrors", true);
        }
    },

    cancel : function(component, event, helper) {
        $A.get("e.force:closeQuickAction").fire();
    }
})
```

CreateUserHelper.js

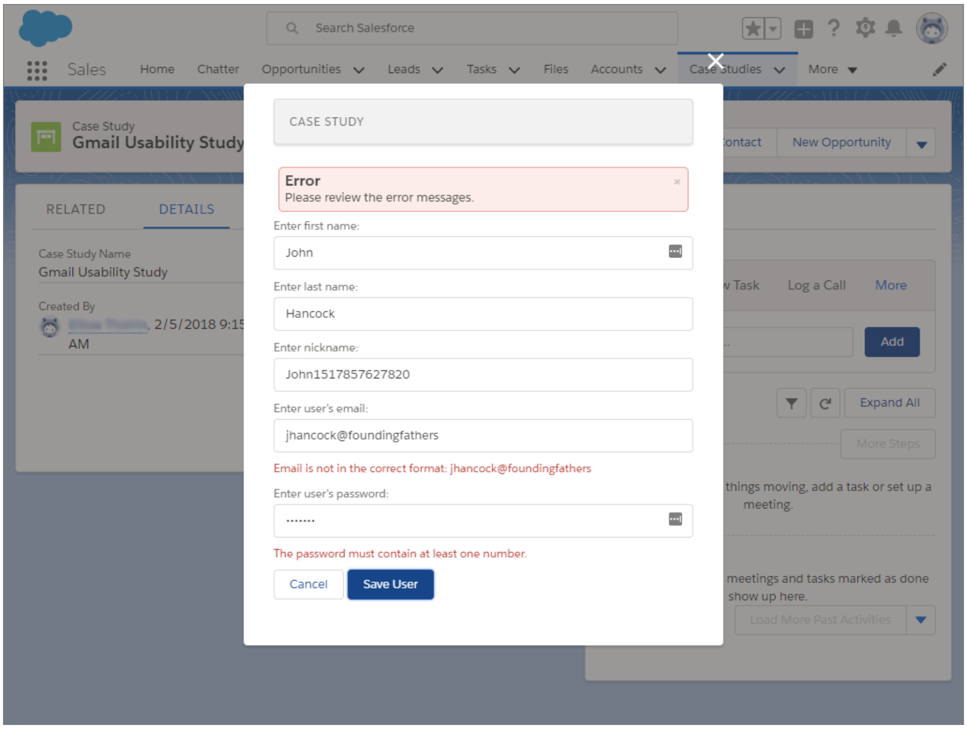
```
({
    save: function(component, name, password, email, nickname) {
        // Create a user record, save it, and close the panel
        var userRecord = {apiName: 'Test_User__c', fields: {}};
        userRecord.fields.Name = {value: name};
        userRecord.fields.Password__c = {value: password};
        userRecord.fields.Email__c = {value: email};
        userRecord.fields.Nickname__c = {value: nickname};
        userRecord.fields.Case_Study__c = {value:
component.get("v.recordId")};
        // get the force:recordData and set the targetRecord
        component.find("frd").set('v.targetRecord', userRecord);
        // invoke saveRecord of force:recordData

component.find("frd").saveRecord($A.getCallback(function(response) {
            if (component.isValid() && response.state == "SUCCESS") {

                $A.get("e.force:closeQuickAction").fire();
                var toastEvent = $A.get("e.force:showToast");
                toastEvent.setParams({
                 "title": "Success!",
                    "message": "The test user has been created."
                });
                toastEvent.fire();
                $A.get('e.force:refreshView').fire();
            } else if (response.state == "ERROR") {
                console.log('There was a problem and the state is: '+
 response.state);
            }
        }));
    },

    validatePassword : function(component, event, helper) {
        var inputCmp = component.find("userPassword");
        var value = inputCmp.get("v.value");

        if (value == undefined) {
           inputCmp.set("v.errors", [{message: "You must enter a
password."}]);
        } else if (value.length < 7 || value.length > 15) {
            inputCmp.set("v.errors", [{message: "The password is the
wrong length (must be <= 15): " + value.length}]);
        } else if (value.search(/[0-9]+/) == -1) {
            inputCmp.set("v.errors", [{message: "The password must
contain at least one number."}]);
        } else if (value.search(/[a-zA-Z]+/) == -1) {
            inputCmp.set("v.errors", [{message: "The password must
contain at least one letter."}]);
        } else {
            inputCmp.set("v.errors", null);
        }
 },

    validateEmail : function(component, event, helper) {
        var inputCmp = component.find("userEmail");
        var value = inputCmp.get("v.value");

        if (value == undefined) {
           inputCmp.set("v.errors", [{message: "You must enter an
email."}]);
           return;
        }

        var apos = value.indexOf("@");
        var dotpos = value.lastIndexOf(".");

        if (apos<1||dotpos-apos<2){
            inputCmp.set("v.errors", [{message: "Email is not in the
correct format: " + value}]);
        } else if (value.substring(apos+1, dotpos) != "gmail") {
           inputCmp.set("v.errors", [{message: "Email must be a gmail
 account: " + value.substring(apos+1, dotpos)}]);
        } else {
            inputCmp.set("v.errors", null);
        }
 }
})
```

After creating the Lightning component, we assign it to an action. In the object management settings for Case Study, we go to Buttons, Links, and Actions, click **New Action**, then configure the action with these parameters.

Field	Value
Object Name	Case Study

Field	**Value**
Action Type	Lightning Component
Lightning Component	c:CreateUser
Height	500px
Label	Create Test User
Name	CreateUser

Then we add our new Lightning action to the Case Study page layout. When users invoke it from a case study record page, they see the Lightning action that we created.



Case Study Name
Gmail Usability Study

Created By
[redacted], 2/5/2018 9:15 AM

things moving, add a task or set up a meeting.

meetings and tasks marked as done show up here.

CASE STUDY

Error
Please review the error messages.

Enter first name:
John

Enter last name:
Hancock

Enter nickname:
John1517857627820

Enter user's email:
jhancock@foundingfathers

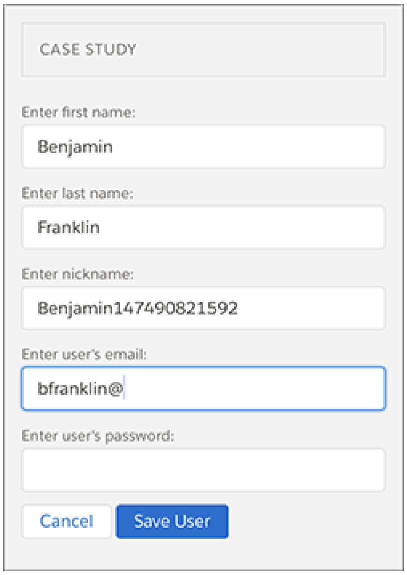
Email is not in the correct format: jhancock@foundingfathers

Enter user's password:
•••••••

The password must contain at least one number.

Cancel | Save User

The great thing about this Lightning action is that it also works in the Salesforce mobile app.


CASE STUDY

Enter first name:
Benjamin

Enter last name:
Franklin

Enter nickname:
Benjamin147490821592

Enter user's email:
bfranklin@

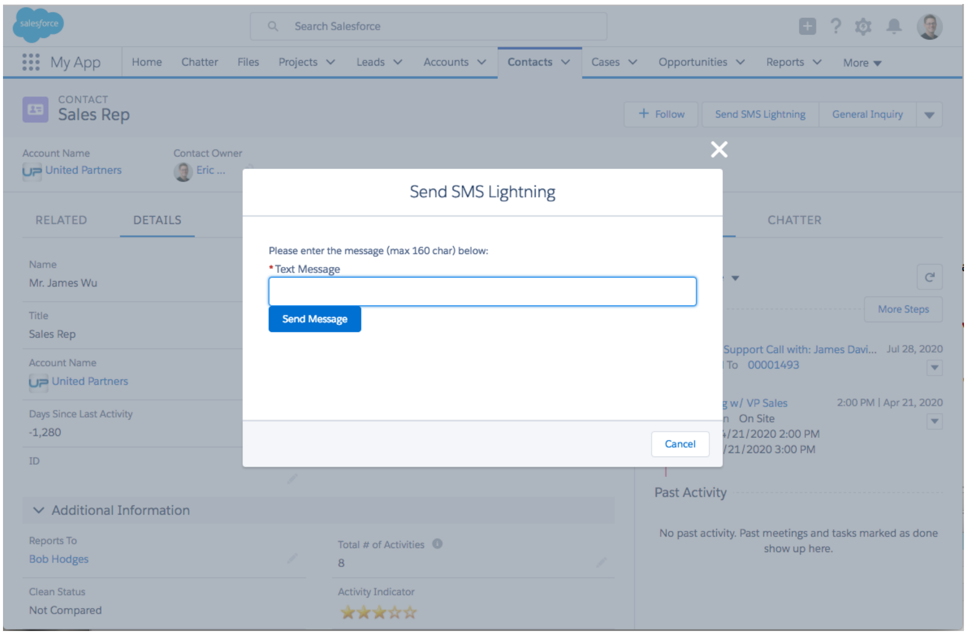
Enter user's password:

Integrate Third-Party APIs

Maybe you use JavaScript buttons for integration with third-party systems. Can you use Lightning actions for that? You sure can. The primary difference is that for integration using Lightning actions, you have to use a server-side controller. In return, you get better handling of security credentials and the ability to use Apex for asynchronous and batch API calls.

Let's look at how to integrate with Twilio for sending SMS messages. In this example, we're in the luxury travel business, and we handle celebrities and VIPs. We want our customer service agents to be able to communicate with their clients, but we don't want to expose the clients' personal contact information to anyone. We create a Lightning action on the Contact object so that agents can send messages without seeing the contact's phone number.

As in our case study example, we'll create the Aura component and helper classes, and then create a quick action to invoke the component.



This Lightning action is composed of the Aura component, a JavaScript controller, and an Apex controller, which references library classes that handle the Twilio integration.

SendTwilioSMS.cmp

```
<aura:component controller="TwilioSendSMSController"
implements="flexipage:availableForAllPageTypes,force:hasRecordId,force:lightningQuickAction"
 >
   <aura:attribute name="textMessage" type="String" />
   <aura:attribute name="destinationNumber" type="String" />
   <aura:attribute name="messageError" type="Boolean" />
   <aura:attribute name="recordId" type="String" />

      <aura:handler name="init" value="{!this}" action="{!c.init}" />


   <aura:if isTrue="{!v.messageError}">
      <!-- Load error -->
      <div class="userCreateError">
         <ui:message title="Error" severity="error" closable="true">
            Unable to send message. Please review your data and try
again.
```

```
         </ui:message>
      </div>
   </aura:if>

   <div class="slds-form--stacked">
     <label class="slds-form-element__label" for="instructMsg">Please
 enter the message (max 160 char) below: </label>
      <br/>
      <div class="slds-form-element__control">
        <ui:inputText class="slds-input" aura:id="message" label="Text
 Message" value="{!v.textMessage}" required="true" maxlength="160"
size="165" />
      </div>
      <div class="centered">
         <ui:button class="slds-button slds-button--brand"
press="{!c.sendMessage}" label="Send Message"/>
      </div>
   </div>
</aura:component>
```

SendTwilioSmsController.js

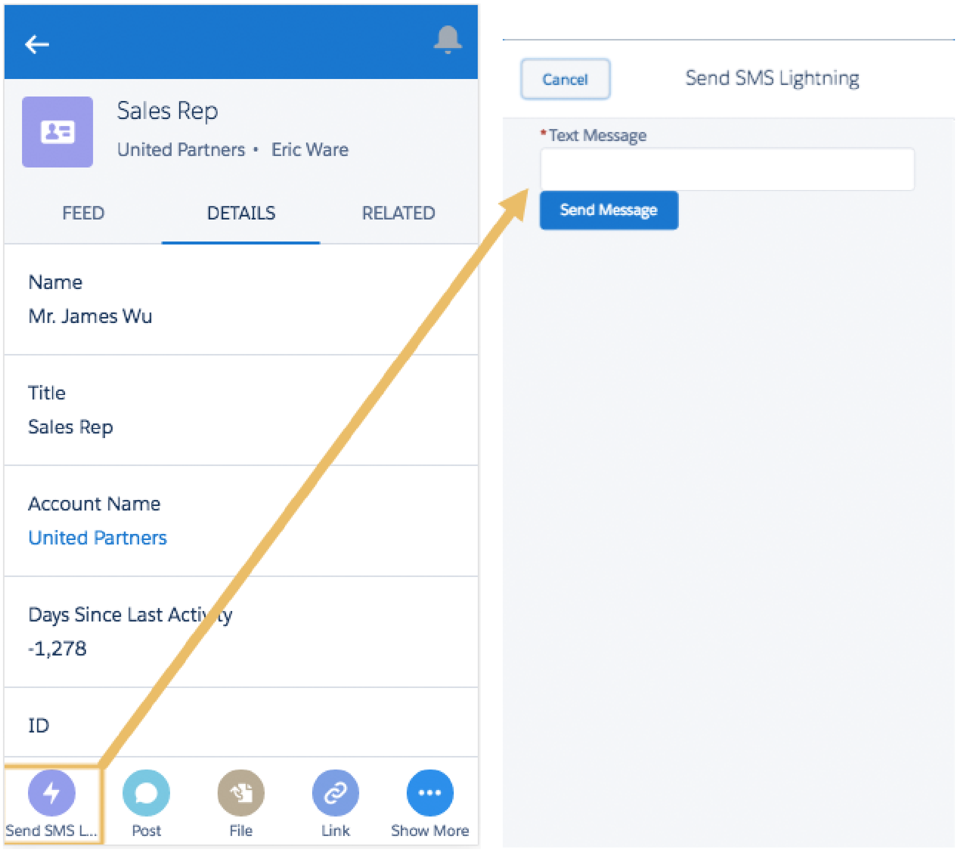
```
({
   init : function(component, event, helper) {
      var action = component.get("c.getPhoneNumber");
      action.setParams({"contactId": component.get("v.recordId")});
      action.setCallback(this, function(response) {
         var state = response.getState();
         if(component.isValid() && state == "SUCCESS"){
            component.set("v.destinationNumber",
response.getReturnValue());
         } else {
            component.set("v.messageError", true);
         }
      });
      $A.enqueueAction(action);
   },

      sendMessage : function(component, event, helper) {
      var smsMessage = component.get("v.textMessage");
      var number = component.get("v.destinationNumber");
      var recordId = component.get("v.recordId")
      var action = component.get("c.sendMessages");
      action.setParams({"mobNumber": number, "message": smsMessage,
"contactId": component.get("v.recordId")});
      action.setCallback(this, function(response) {
         var state = response.getState();
         if(component.isValid() && state == "SUCCESS"){
            $A.get("e.force:closeQuickAction").fire();
            var toastEvent = $A.get("e.force:showToast");
            toastEvent.setParams({
               "title": "Success!",
               "message": "SMS has been sent woo hoo!"
            });
            toastEvent.fire();
         } else {
            component.set("v.messageError", true);
         }
      });
      $A.enqueueAction(action);
   }
})
```

SendTwilioSmsController.apxc

```
/*
* Apex controller that currently contains only one method to send sms
 message
*/
global class TwilioSendSMSController {

   /*
   * This method uses the Twilio for Salesforce library class and
method to
   * send the message using the Twilio api.
   */
   @AuraEnabled
      webService static String sendMessages(String mobNumber, String
message, Id contactId) {
         System.debug('the mobNumber is: '+ mobNumber + ' and the
message is: '+ message + ' and contactId is: ' + contactId);

         if (mobNumber == null) {
            mobNumber = getPhoneNumber(contactId);
         }

         try {
            TwilioRestClient client = TwilioAPI.getDefaultClient();

            Map<String,String> params = new Map<String,String> {
               'To' => mobNumber,
               'From' => '15555551234',
               'Body' => message
               };
            TwilioSMS sms =
client.getAccount().getSMSMessages().create(params);
            return sms.getStatus();
         } catch(exception ex) {
            System.debug('oh no, it failed: '+ex);
            return 'failed';
         }
      }

      @AuraEnabled
      public static String getPhoneNumber(Id contactId) {
         Contact currentRecord = [SELECT Phone FROM Contact WHERE Id
= :contactId];
         return currentRecord.Phone.replace(' ', '').replace('-',
'').replace(')', '').replace('(', '');
    }
}
```

After creating the component, we create a Lightning action. This time we add it to the Contact page layout so agents can access it.

Again, the cool thing is that the action is also available in the Salesforce mobile app. If a car service provider is trying to contact the customer at the airport, the driver can use a mobile device to reach the customer easily.



Lightning actions are the future of programmatic actions in Lightning Experience. We hope that you start looking at this set of solutions as a better alternative to JavaScript buttons.

And if you depend on partner apps that use JavaScript buttons, you'll be glad to know that many of our partners have already started migrating and upgrading their apps to Lightning. You can find more apps on the AppExchange that are updated for Lightning Experience.

Beyond the Basics

If you'd like to try Twilio for yourself, you can find an unmanaged package of the Twilio for Salesforce library class used in the third-party integration example in the Salesforce Github library: https://github.com/twilio/twilio-salesforce

Before using Twilio for Salesforce and the Twilio API, create a Twilio account, set up a phone number, and connect your account to Salesforce. See the `TwilioApi.cls` in the Github library for more detailed instructions.

MORE ABOUT VISUALFORCE & LIGHTNING EXPERIENCE

CHAPTER 34 Use Visualforce in Lightning Experience

In this chapter ...

Lightning Experience brings an all new user interface to your Salesforce organization, but that doesn't mean your Visualforce apps stop working. Visualforce pages work in Lightning Experience, many without any revisions. Things have moved around, though, and there are some chores you'll want to complete to make sure your Visualforce pages work the way you expect as your users switch between Lightning Experience and Salesforce Classic. And there are a very few features that, alas, don't work in Lightning Experience. We'll get you sorted on all of it in this module.

Let's start with a few basic details. These are topics we'll cover in depth later, but let's handle some essential items right up front.

- With some important exceptions, Visualforce "just works" in Lightning Experience. If you've written Visualforce apps for your organization, you can expect that they work whether your users access them in Lightning Experience or Salesforce Classic.
- If your Visualforce pages use the built-in standard components, their look-and-feel matches Salesforce Classic, whether your users access them in Lightning Experience or Salesforce Classic. If you want your pages to match the Lightning Experience styling, you have some work to do.
- If your Visualforce pages make use of JavaScript, there are things you need to check. Visualforce doesn't "own" the whole page when shown in Lightning Experience, and because of that your JavaScript code needs to play by some new rules.
- There are other things that have changed about how Visualforce runs when it's running inside Lightning Experience. For the most part, these are turning the "just works" crank, but you'll want to be aware of them all the same.

And finally, did we mention that some things have moved around? Have they ever! Lightning Experience is a complete rethinking of how to use Salesforce, and while the job's not done yet, we're really excited about where we’re going. To get you oriented for where your Visualforce is in the new environment, let’s take a quick tour of some of the places you can use Visualforce in Lightning Experience.

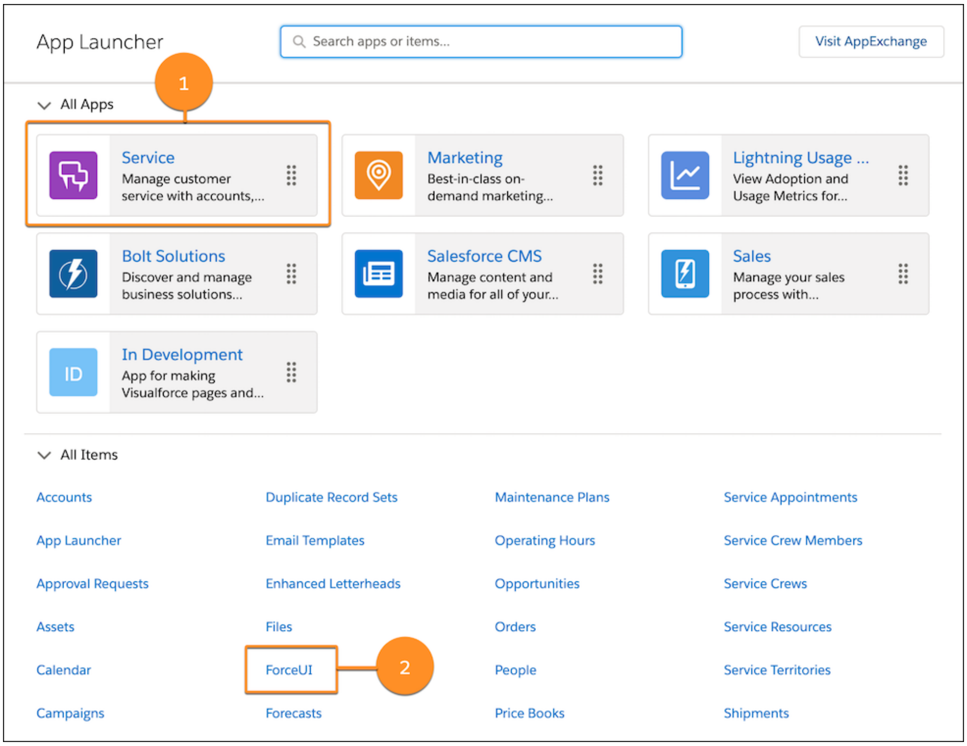
Where You Can Use Visualforce in Lightning Experience

As with Salesforce Classic, you can extend Lightning Experience with your custom Visualforce pages and apps. But where you find them has changed, and there are still some places you can't put Visualforce.

The following are some of the ways you can add Visualforce to your Lightning Experience organization. This is just a quick tour, though. For more details on how to customize your organization using Visualforce pages, see the resources at the end of this unit.

Example: **Open a Visualforce Page from the App Launcher**

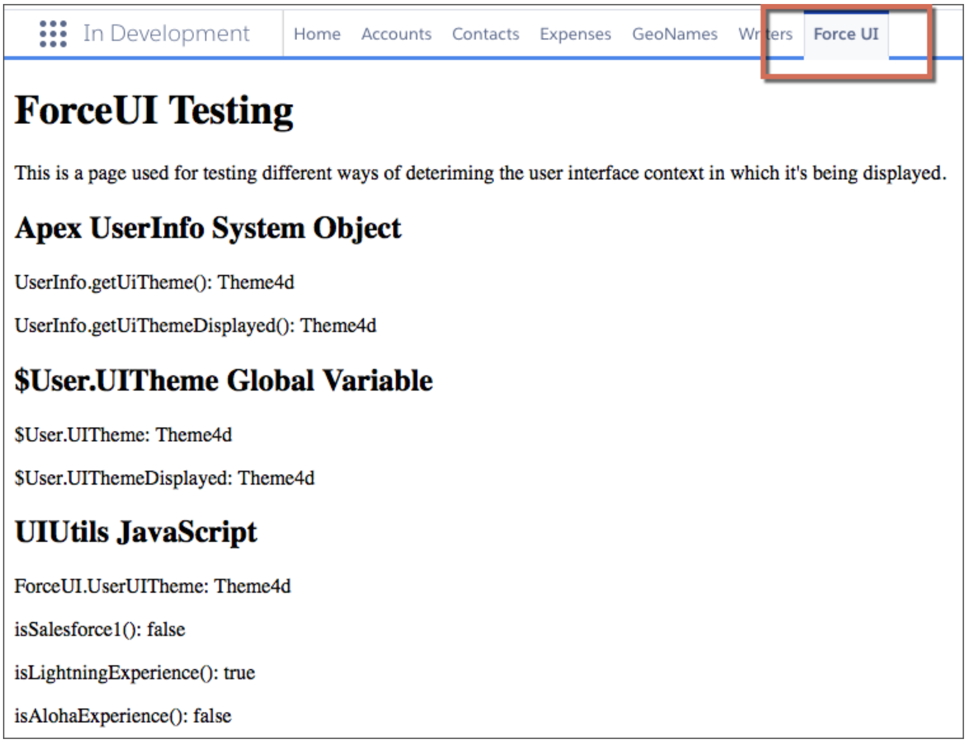
Your Visualforce apps and custom tabs are all available from the App Launcher. To open the App Launcher, click ⠿ in the navigation bar. To see all of your apps and items, select **View All**.



Click a custom app (1) to activate it. Items in the app display in the navigation bar, including any Visualforce tabs you've added to the app. Note that you need to add your Visualforce pages to tabs for them to be accessible in the App Launcher. Visualforce tabs that aren't in apps can be found in All Items (2).

Example: **Add a Visualforce Page to the Navigation Bar**

As described in the preceding example, you can add Visualforce tabs to an app and they display as items in the app's navigation bar.



ForceUI Testing

This is a page used for testing different ways of deteriming the user interface context in which it's being displayed.

Apex UserInfo System Object

UserInfo.getUiTheme(): Theme4d

UserInfo.getUiThemeDisplayed(): Theme4d

$User.UITheme Global Variable

$User.UITheme: Theme4d

$User.UIThemeDisplayed: Theme4d

UIUtils JavaScript

ForceUI.UserUITheme: Theme4d

isSalesforce1(): false

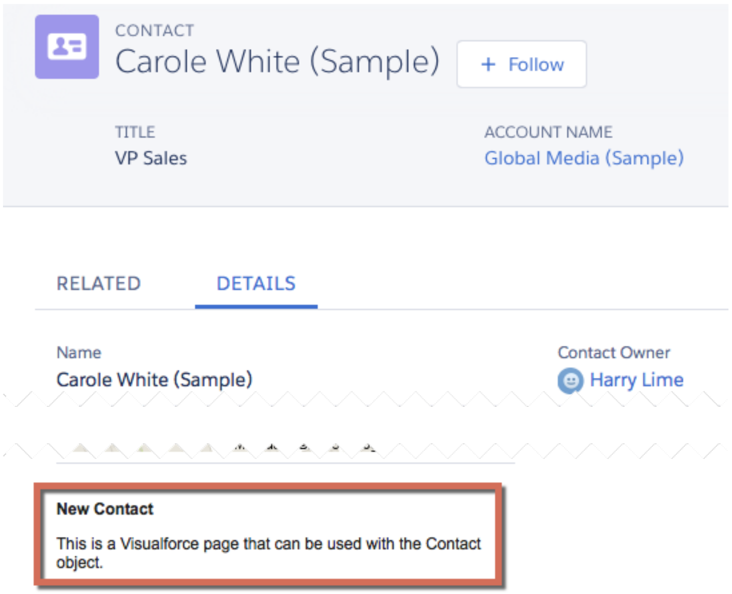
isLightningExperience(): true

isAlohaExperience(): false

(And hey, does a "ForceUI" utility page sound interesting? Keep reading this module!)

Example: **Display a Visualforce Page within a Standard Page Layout**

Extend your page layouts by embedding Visualforce pages on them to display completely custom content on a standard page. The behavior here is identical to Salesforce Classic, except you need to view the record's Details to see the page layout.

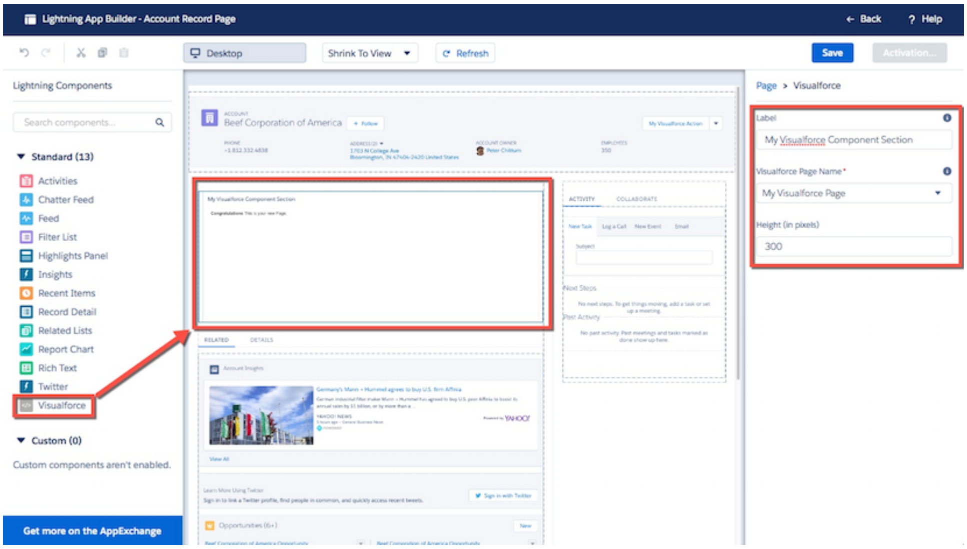


New Contact

This is a Visualforce page that can be used with the Contact object.

Example: Add a Visualforce Page as a Component in the Lightning App Builder

When you create a custom app page in the Lightning App Builder, you can add a Visualforce page to the page by using the Visualforce component.



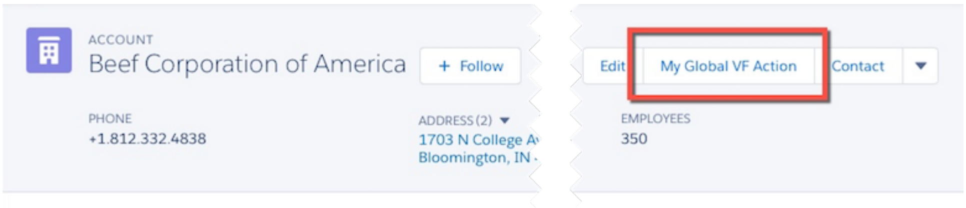
Label: My Visualforce Component Section

Visualforce Page Name*: My Visualforce Page

Height (in pixels): 300

> **Note:** You must enable `"Available for Lightning Experience, Lightning Communities, and the mobile app"` for a Visualforce page to make it available in the Lightning App Builder.

Example: Launch a Visualforce Page as a Quick Action

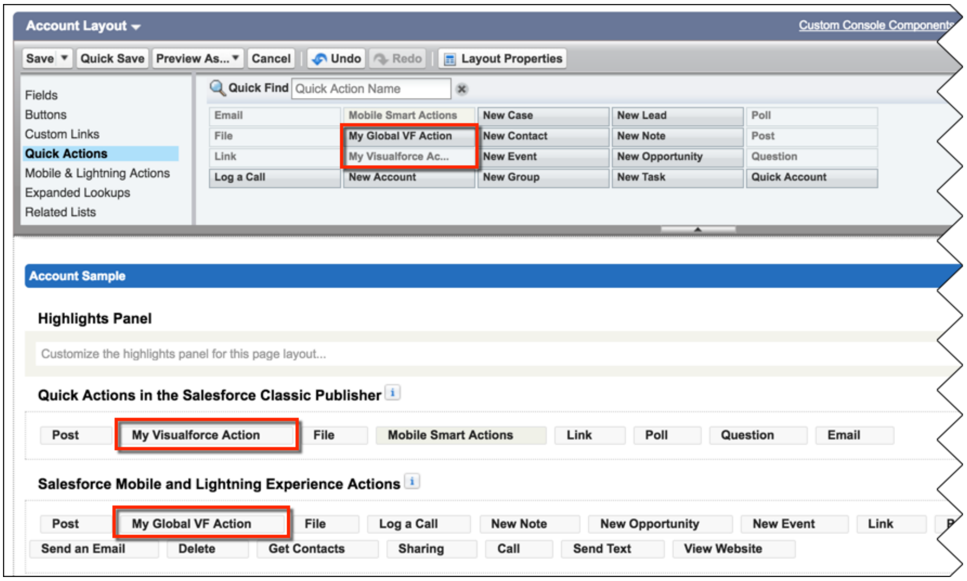


PHONE +1.812.332.4838

ADDRESS (2) 1703 N College A Bloomington, IN

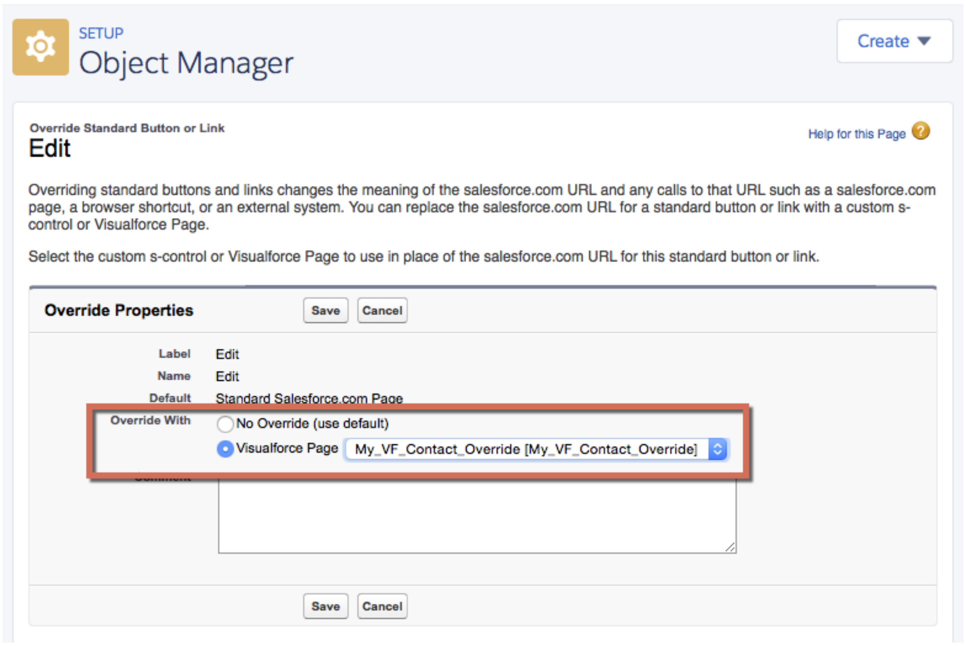
EMPLOYEES 350

Although their placement in the Lightning Experience user interface is quite different from Salesforce Classic, the process of adding quick actions is much the same. Add them to the appropriate publisher area on the object's page layout.



Example: **Display a Visualforce Page by Overriding Standard Buttons or Links**

You can override the actions available on an object with a Visualforce page. When the user clicks a button or link that has been overridden, your page displays instead of the standard page. Setting this up is pretty much identical to Salesforce Classic. Indeed, you'll have a hard time telling that you're in Lightning Experience when defining an action override!



Override Standard Button or Link

Edit

Overriding standard buttons and links changes the meaning of the salesforce.com URL and any calls to that URL such as a salesforce.com page, a browser shortcut, or an external system. You can replace the salesforce.com URL for a standard button or link with a custom s-control or Visualforce Page.

Select the custom s-control or Visualforce Page to use in place of the salesforce.com URL for this standard button or link.

Override Properties

Label	Edit
Name	Edit
Default	Standard Salesforce.com Page
Override With	No Override (use default) [✓ Visualforce Page] My_VF_Contact_Override [My_VF_Contact_Override]
Comment	

Example: Display a Visualforce Page Using Custom Buttons or Links

You can create new actions for your objects, in the form of buttons and links, by defining them on an object. JavaScript buttons and links aren't supported in Lightning Experience, but Visualforce (and URL) items are. The process of defining Visualforce buttons and links is identical to that in Salesforce Classic, so we won't bother to show it here.

CHAPTER 35 Developing Visualforce Pages for Lightning Experience

In this chapter ...

The development process for creating Visualforce pages and apps for Lightning Experience is in some ways considerably different from developing for Salesforce Classic. In others, you'll find it's just the same. The main difference is how you view and test your pages during development.

In this unit we'll cover the details of getting your development environment set up, and then get into the details of the "right" way to test your pages while you're in the process of building them. The good news is that the process you need to use to develop for Lightning Experience is the same you'll use for developing Salesforce mobile pages as well.

Set Up Your Editor

The first thing you'll want to set up is the editing tool you'll use for writing code. This process remains the same, whether you're creating pages for Lightning Experience, Salesforce Classic, or the Salesforce mobile app, and whether you're using the Developer Console, the Salesforce Extensions for Visual Studio Code, or the good old Setup editor.

If you've already got a preferred Visualforce editing tool, you don't need to do anything here. Writing and saving your Visualforce markup remains exactly the same. The Developer Console has its own user interface, and doesn't change between Lightning Experience and Salesforce Classic. The editor in Setup is also unchanged, retaining the Salesforce Classic user interface in all user interface contexts. And of course if you're using a native tool, such as the Salesforce Extensions for Visual Studio Code or one of the many third-party tools available, those have their own user interfaces.

The one exception is the editor in the Visualforce Development Mode footer. If you've enabled Development Mode in your user settings, and you're using Salesforce Classic, then viewing and editing Visualforce pages with the Development Mode footer is unchanged, as you'd expect. If you switch to Lightning Experience, and then access a page using the traditional `https://`*`yourInstance`*`.salesforce.com/apex/`*`PageName`* URL pattern, you might be somewhat surprised to find yourself back in Salesforce Classic.

This is expected, and we'll talk about it more when we get to viewing and testing your Visualforce pages. For now, know that Development Mode for Visualforce is only available in Salesforce Classic.

Viewing Visualforce Pages During Development

Viewing your Visualforce pages while they're being developed is a common task. And while it's not "testing" in the formal sense, you certainly want to be able to interact with functionality you've built to ensure it's making progress towards correct behavior. This is frequently accomplished by accessing the page using the `https://`*`yourInstance`*`.salesforce.com/apex/`*`PageName`* URL pattern. While this still works for reviewing pages in Salesforce Classic, it doesn't work for checking behavior in Lightning Experience.

Pages you view using direct URL access always display in Salesforce Classic, which is to say, the "classic" Visualforce container, no matter what your user interface settings are. If you create Visualforce pages that have Lightning Experience-specific behavior, you can't review that behavior just by using the usual direct URL access.

Beyond the Basics

What's going on behind the scenes that causes this? It's pretty simple, really. In order to view your page in Lightning Experience, you need to access the Lightning Experience container app. This means accessing `/lightning`. If you're accessing that, you can't access `/apex/`*`PageName`*. They're just two different URLs that don't overlap.

So what's a developer to do? You need to view your page from within the Lightning Experience app itself, so that it's running inside the Lightning Experience container. This means you'll need to navigate to the page in Lightning Experience, and there are a variety of ways to do that.

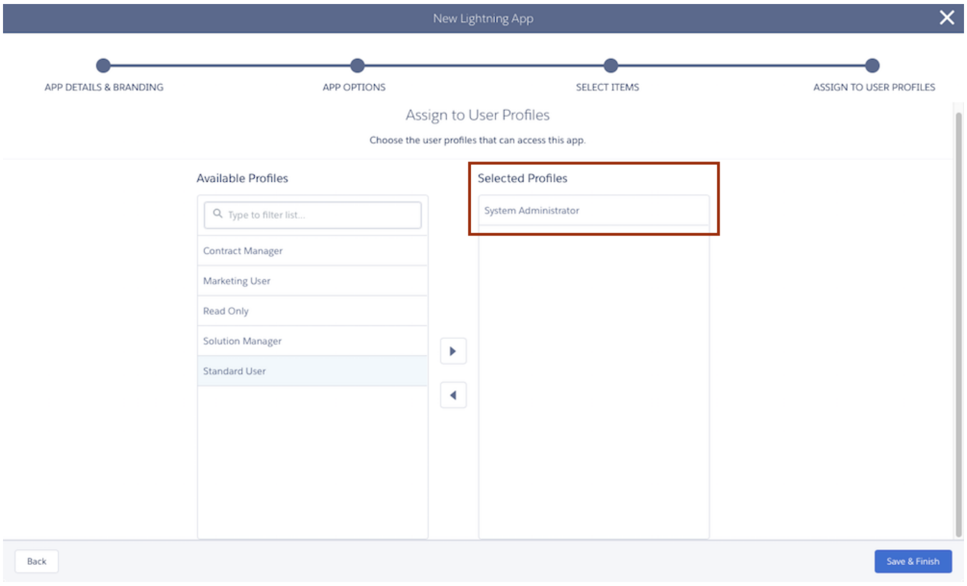
The simplest way to get to a specific Visualforce page is to create a tab for it, and then navigate to that tab via the All Items section in the App Launcher. A more long-term approach would be to create an "In Development" app, and add your Visualforce tabs to it as you work on them, and move or remove them as they roll out to production. Since the controls for doing this have moved around a bit, here are brief instructions.

1. From Setup, enter *`Apps`* in the `Quick Find` box, then select **App Manager**.

   You should see the `Lightning Experience App Manager` Setup page.

2. Click **New Lightning App**, and then create a custom app for your pages in development.

   Consider restricting your app to only System Administrators, or a profile you've created for developers in your organization.



Assign to User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

Contract Manager

Marketing User

Read Only

Solution Manager

Standard User

Selected Profiles

System Administrator

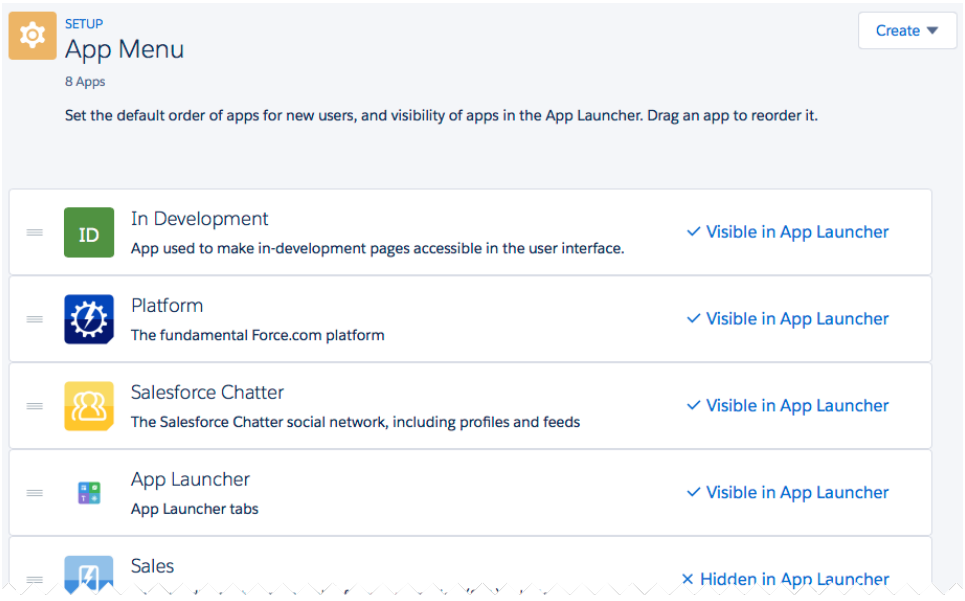
You don't need your users to see your pages before they're added to their permanent place in the user interface.

3. From Setup, enter `App Menu` in the `Quick Find` box, then select **App Menu**.

   You should see the `App Menu` Setup page.

4. Make sure that your In Development app is set to `Visible in App Launcher`.

   While you're at it, you might want to rearrange items, and even hide apps you don't use.



App Menu

8 Apps

Set the default order of apps for new users, and visibility of apps in the App Launcher. Drag an app to reorder it.

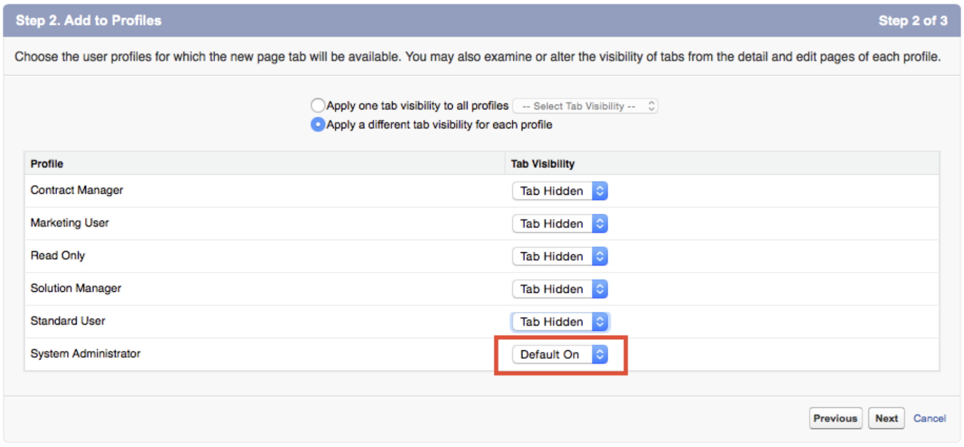
In Development	App used to make in-development pages accessible in the user interface.	✓ Visible in App Launcher
Platform	The fundamental Force.com platform	✓ Visible in App Launcher
Salesforce Chatter	The Salesforce Chatter social network, including profiles and feeds	✓ Visible in App Launcher
App Launcher	App Launcher tabs	✓ Visible in App Launcher
Sales		✕ Hidden in App Launcher

5. From Setup, enter `Tabs` in the `Quick Find` box, then select **Tabs**.

   You should see the `Custom Tabs` Setup page.

6. Click **New** in the Visualforce Tabs section, and then create a custom tab for the page currently in development.

   Make the tab visible only to your development user profile, and add the tab only to your In Development app.

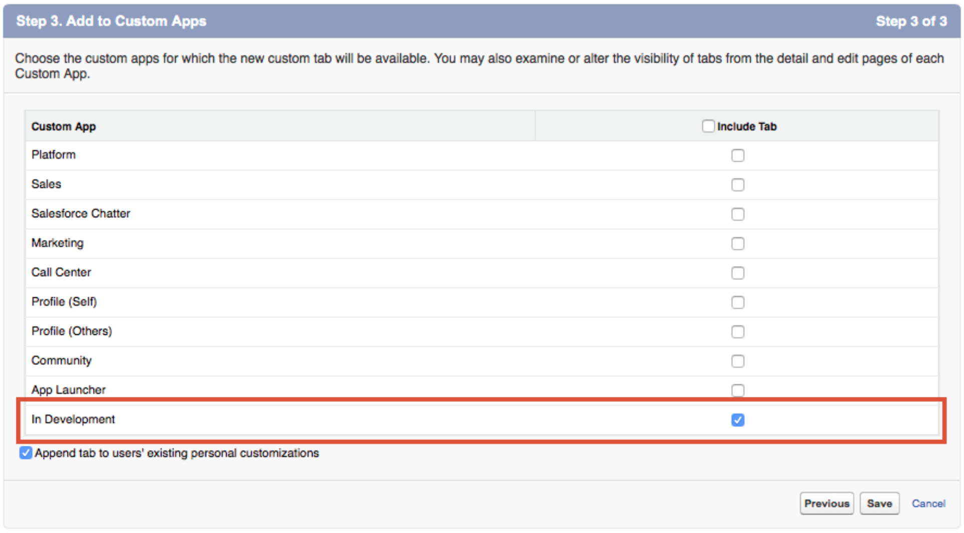


Step 2. Add to Profiles Step 2 of 3

Choose the user profiles for which the new page tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each profile.

Apply one tab visibility to all profiles [-- Select Tab Visibility --]
[✓ Apply a different tab visibility for each profile]

Profile	Tab Visibility
Contract Manager	Tab Hidden
Marketing User	Tab Hidden
Read Only	Tab Hidden
Solution Manager	Tab Hidden
Standard User	Tab Hidden
System Administrator	Default On



Step 3. Add to Custom Apps Step 3 of 3

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.

Custom App	Include Tab
Platform	☐
Sales	☐
Salesforce Chatter	☐
Marketing	☐
Call Center	☐
Profile (Self)	☐
Profile (Others)	☐
Community	☐
App Launcher	☐
In Development	✓

✓ Append tab to users' existing personal customizations

7. Repeat the previous step for each page you want to add to your In Development app. For adding new pages in the future, this is the only step required.

For all that that's an easy way to see your pages while you're working on them, it doesn't really compare to simply typing the page name into a URL. For a similarly low-overhead way to test your page in Lightning Experience, you can type the following into your JavaScript console:

```
$A.get("e.force:navigateToURL").setParams(
    {"url": "/apex/pageName"}).fire();
```

This JavaScript fires the Lightning Experience `navigateToURL` event, and is the equivalent of entering in the classic `/apex/`*`PageName`* URL—you can even see that URL pattern in the code.

Note: You need to currently be in Lightning Experience for this technique to work. If you're in Salesforce Classic, the JavaScript code fails.

For something a little more convenient to use, add the following bookmarklet to your browser's menu or toolbar. (We've wrapped this code for readability.)

```
javascript:(function(){
    var pageName = prompt('Visualforce page name:');
    $A.get("e.force:navigateToURL").setParams(
        {"url": "/apex/" + pageName}).fire();})();
```

This bookmarklet prompts you for a page name, and then fires the event to navigate directly to it. Useful!

Once you've navigated to the page you're working on, you can simply use your browser's reload command to refresh the page as you make changes.

Reviewing Visualforce Pages in Multiple Environments

If you're creating pages that will be used in Lightning Experience, Salesforce Classic, and the Salesforce mobile app, you'll want to be able to review them in all environments while you're working on them. To do so, you'll need to open the page in multiple browsers and on multiple devices.

A Visualforce page that's going to be used across the different Salesforce user interface contexts and form factors is tricky to review while you're in development. You can toggle back and forth between Salesforce Classic and Lightning Experience using the environment selector in the profile menu, but that's going to get old quickly. And you can similarly play with your browser's user agent settings to simulate the Salesforce mobile environment, but that's even more cumbersome.

Instead, you're going to want to use multiple browsers, or even multiple devices, to view your pages. And you'll want to have access to at least one additional test *user* as well. Here's an example of how you might set up your development environment.

Main Development Environment

This environment is where you work in Setup to make changes to your organization, like adding custom objects and fields, and maybe where you write actual code, if you use the Developer Console.

- **Browser**: Chrome
- **User**: Your developer user
- **User interface setting**: Salesforce Classic

Review your page's design and behavior in Salesforce Classic in this environment.

Lightning Experience Review Environment

This environment is where you check your page's design and behavior in Lightning Experience.

- **Browser**: Safari or Firefox
- **User**: Your test user
- **User interface setting**: Lightning Experience

Salesforce Mobile Review Environment

This environment is for checking your page's design and behavior in the Salesforce mobile app.

- **Device**: iOS or Android phone or tablet
- **Browser**: Salesforce app
- **User**: Your test user
- **User interface setting**: Lightning Experience

Note: This is just an example setup, and you can use any modern browsers or mobile devices in yours, for both Salesforce Classic and Lightning Experience. The key is to use two different browsers so you can access both Salesforce Classic and Lightning Experience at once, and to use a real device to test with the Salesforce mobile app.

This probably sounds pretty elaborate, and it *is* a bit of a chore to set up initially, especially if you're champing at the bit to get coding. But keep in mind two things. First, once you're set up, it's done. And second, this workspace doesn't just give you a great development environment, it also provides you with the environments you need for formal testing of your pages. Because you wouldn't dream of putting your pages into production without formal testing, right?

Testing Your Visualforce Pages

Testing your Visualforce pages before deploying them into production is an essential development task. When your organization adopts Lightning Experience the process of testing your pages becomes more complicated.

We just talked about the environments you need to set up for doing quick, informal testing while you're developing your pages. The need for those environments also applies to testing your pages and apps formally. Rather than repeating them, let's talk a bit about *why* you need these different environments.

The need to test your pages in both Lightning Experience and Salesforce Classic is fairly obvious, but why can't you do that testing in the same browser, with the same user? In fact, you can and you should. Your users can toggle back and forth between the different user interfaces, and you should verify that your pages work when they do so.

But you also want to test pages in a more isolated and systematic fashion, so that you're sure that what you're testing is the page's basic functionality, as separated as possible from the effects of other code, whether that code is yours, ours, the browser's, or the device's.

Desktop and mobile browsers, even from the same vendor, behave differently—sometimes *very* differently. You cannot do rigorous, formal testing unless you're testing on every device and every browser you plan to support.

If you're developing an individual page, or a basic app for identical devices, your "matrix" of the different factors might be simple. But for more ambitious projects, if you're developing functionality that you need to support across a range of possibilities, your test plan should take into consideration the need to test across:

- Each different supported device.
- Each different supported operating system.
- Each different supported browser—including the Salesforce mobile app, which embeds its own.
- Each different supported user interface context (Lightning Experience, Salesforce Classic, and the Salesforce mobile app).

Fortunately for your sanity, some of these factors collapse together, reducing the combinatorial explosion. For example, most Apple mobile devices can be counted on to be updated to the latest version of iOS. This means that the device, operating system, and browser are effectively only one combination. Your test plan might therefore choose to test only one iPhone and one iPad, updated to the latest iOS and Salesforce app.

A final word about testing. Another reason we strongly suggest your development and test environments be similar is so that you can start testing, full testing, early in the development process. We've found that it's all too easy to put testing on secondary devices off until late in a project. When that happens it's inevitably a setback when problems are discovered.

Test early, test often, test everything.

CHAPTER 36 Exploring the Visualforce App Container

In this chapter ...

The largest difference between Visualforce in Lightning Experience and Visualforce in Salesforce Classic is the environment it runs in. In Salesforce Classic, Visualforce "owns" the page, the request, the environment. Visualforce *is* the application container. But in Lightning Experience, Visualforce runs inside an iframe that's wrapped inside the larger Lightning Experience container.

This change to the execution context has a number of effects on the way Visualforce pages can affect the overall Salesforce application. We'll talk about these changes in this unit, but save the full details of a few of them for their own units.

Note: This unit is a little more "under construction" than the rest. The reason is simple: The impact of the issues described here is highly dependent on your code. We've worked really hard to make things "just work" for you, and in most cases little or nothing here will show on your radar. But we can't anticipate every way that you're using Visualforce. Here we're outlining the general aspects of how Lightning Experience affects Visualforce. When you have conversations with us, and as we learn more from you about actual impact, we can offer explanations with more details about how to address specific issues.

The Outer Lightning Experience Container

Let's start with the outer container, the Lightning Experience application. The Lightning Experience container is a "single-page application," or SPA, which is accessed at the `/lightning` URL. The `/lightning` page loads, its code starts up, and that application code takes over the environment.

The process by which a single-page application loads its resources—usually a static HTML shell and a lot of JavaScript—is both interesting and complex. If you've worked with JavaScript frameworks like AngularJS or React, you're reasonably familiar with the basics of how Lightning Experience, in the form of `/lightning`, starts up. And to be honest, the full details don't matter. You don't have any control over it, and the implementation continues to evolve.

Here's what's important to know: Lightning Experience, or `/lightning`, is in charge of the request. Your Visualforce page is not. Your page needs to work within constraints that Lightning Experience imposes upon it. Lightning Experience is the parent context, and your Visualforce page is the child context. Children need to obey their parents.

Some of these constraints, such as the size of the frame in which your Visualforce page is displayed, are imposed directly by Lightning Experience. They're easier to understand and work with, and we'll talk about them in a minute.

Other constraints are implicit, and enforced not by Lightning Experience but by the browser running it. These are mostly security and JavaScript execution constraints. Most pages aren't impacted by these security constraints, and those that are usually fail early and with clear error messages. JavaScript errors are harder to discover and diagnose, but there are some general rules we'll cover in a bit.

The Visualforce iframe

When your Visualforce page runs in Lightning Experience, it's displayed inside an HTML iframe. An iframe creates an embedded browsing context that's effectively a separate browser "window" from the main Lightning Experience browsing context. The iframe creates a boundary between the Visualforce page and its parent, the Lightning Experience application.

The advantage of running Visualforce pages inside an iframe is that, for pages that don't need to access or change the top-level browsing context, running inside the iframe looks almost exactly like running as a page in Salesforce Classic. This is why you don't need to modify all of your Visualforce pages to adapt to the wildly different behind-the-scenes request environment of Lightning Experience. It's an important part of the "just works" strategy for supporting Visualforce.

Of course, the flip side is that pages that *do* need to access the top-level browsing context, well, there's some things that need to change. We'll cover some specifics in the next section.

If your page is communicating with services besides Salesforce, the iframe boundary might also result in you needing to update your organization's CORS settings, remote site settings, clickjack settings, or content security policy. Since these depend on security policies and settings outside of Salesforce, we can't provide a recipe for specific changes. We simply call it to your attention here.

Impact of the New Container

The effects of the new Visualforce container—embedding the Visualforce page into an iframe within the Lightning Experience app—can be broadly divided into two categories, which we'll call *security* and *scope*.

Again, we want to emphasize: many, or even most Visualforce pages won't be affected by these issues. But for those that are, we're thinking "forewarned is forearmed." You'll find the source of the problem faster if we've already talked about it together.

Security Impact

Elements of security that might be affected include the following.

- Session maintenance and renewal
- Authentication
- Cross-domain requests
- Embedding restrictions

We discussed a few of these briefly already, the items dealing with cross-domain requests. That is, when the content in the full browser window comes from requests to different servers and services, there's the potential for any of those requests to balk at being displayed in a context that it's not prepared for. Your mission, should the need arise, is to prepare those services to handle requests intended to be put together within the Lightning Experience context. As we said before, the details vary, so we can't provide specific answers here.

One thing we do want to mention specifically is session maintenance. A "session" for our purposes here is basically some kind of token that your browser re-uses from request to request so that you don't need to enter your username and password for every request. You often need to access the current session using the global variable `$Api.Session_ID`.

Here's the thing to keep in mind. `$Api.Session_ID` returns different values depending on the domain of the request. This is because the session ID varies during a session whenever you cross a hostname boundary, such as `.salesforce.com` to `.visual.force.com`. Normally Salesforce transparently handles session hand-off between domains, but if you're passing the session ID around yourself, be aware that you might need to re-access `$Api.Session_ID` from the right domain to ensure a valid session ID.

Lightning Experience and Visualforce pages are not only held in different browser contexts, they're also served from different domains. So, even though it's all showing in one browser window, the session ID *inside* the Visualforce iframe will be different than the session ID *outside* the iframe, in another part of Lightning Experience. Salesforce and Lightning Experience handle this transparently in normal use. But if you're passing around the session ID like hors d'oeuvre at a party (not usually a good idea), you might need to review how you're handling it.

Scope Impact

When we talk about scope we're mainly talking about the following kinds of things.

- DOM access and modification
- JavaScript scope, visibility, and access
- JavaScript global variables such as `window.location`

If this list sounds complicated or confusing, don't worry, we can boil it down to something simple and easy to remember: Don't touch someone else's stuff. Specifically, your JavaScript code (and stylesheet rules, for that matter) can affect elements—DOM nodes, JavaScript variables, and so on—in your page's browsing context, but it can't access elements in any other browsing context, like the parent Lightning Experience context. Don't touch other contexts' stuff!

Practically speaking, the most common code pattern where you'd want to do this kind of thing is to manipulate `window.location` to navigate to another page. This is such a common thing to do, we've written up details on this specific issue...well, by the time you're done with this module, you'll be sick of hearing about it, we promise.

One last note. If you're an experienced JavaScript developer, you're probably already thinking you know how to deal with "I don't have access to the parent browsing context" issues, by using `contentWindow`, `window.parent`, or the like. Please don't. You'll likely run afoul of the same-origin policy (Visualforce and Lightning Experience are served from different domains, remember?). Even if you don't, you're probably replacing obvious, blocking bugs with subtle, intermittent bugs. Where do you want to spend your time: Doing things right, or the debugger?

Doing things right means calling APIs we've made available in your Visualforce pages, primarily for navigation. If you really need to affect things across frame boundaries, use `window.postMessage` to send a message to receiving code in the other frame.

Visualforce Defaults and Environment Changes in Lightning Experience

When your Visualforce pages run in Lightning Experience a number of low-level changes happen behind the scenes. These changes enable most pages to “just work” in the Lightning Experience container, and sometimes you can just be happy they’re there. But you’ll still want to know they’re happening, especially when you’re working on advanced application flows, or troubleshooting a tricky problem.

Some of these changes are simple, and obvious once you think about them. For example, Visualforce pages that run in Lightning Experience always have the standard Salesforce Classic header and sidebar suppressed. Other changes aren’t as visible, but have just as large an impact.

`<apex:page>` `showHeader` and `sidebar` Attributes Are Always `false`

These attributes affect the Salesforce Classic header and sidebar on Visualforce pages. The Salesforce Classic header and sidebar are always suppressed when pages run in Lightning Experience, in favor of Lightning Experience navigation elements. There are no corresponding attributes to affect the Lightning Experience header or sidebar because they can’t be suppressed.

If your page is shared between Salesforce Classic and Lightning Experience, you can still set these attributes to the values you’d like to use when the page runs in Salesforce Classic.

Note: The `standardStylesheets` attribute of `<apex:page>`, which determines whether to include or suppress the standard Salesforce Classic stylesheets, is unaffected by Lightning Experience. That is, it defaults to `true` in Lightning Experience, but you’re able to change it.

The `sforce.one` JavaScript Utility Object

Although `sforce.one` sounds like a droid working in the Salesforce cantina,* it’s actually a utility object that provides a number of useful functions you can use in your own JavaScript code.

`sforce.one` is automatically injected into your page when it runs in Lightning Experience or the Salesforce app. You’ll see it in your JavaScript debugger console and web developer resources list. There’s nothing you need to do to add it, and there’s no way to suppress it, either. (Sadly, there’s no way to get `sforce.one` in your Visualforce pages in Salesforce Classic.)

`sforce.one` is primarily used to fire navigation events. The full details are in an upcoming unit, Managing Navigation.

* There is no Salesforce cantina. Alas.

CHAPTER 37 Sharing Visualforce Pages Between Classic and Lightning Experience

In this chapter ...

We recommend that, wherever possible, you create Visualforce pages that behave correctly whether they run in Salesforce Classic or Lightning Experience. The benefits in terms of reduced complexity in your organization's code and configuration are obvious. And there are a number of contexts, such as Visualforce overrides of standard actions, where you don't have a choice. An action override always uses the same page, whether you're running in Salesforce Classic, Lightning Experience, or the Salesforce app.

It's perfectly reasonable, though, to want slightly or significantly different behavior or styling that's based on the user experience context in which the page is running. In this unit we'll look at a variety of ways to create pages that work correctly in all user experiences, and how your code can detect and make changes for specific contexts.

Detecting and Responding to the User Experience Context in Visualforce Markup

Use the `$User.UITheme` and `$User.UIThemeDisplayed` global variables to determine the current user experience context. You can use these variables in Visualforce expressions to adapt your pages to Lightning Experience, Salesforce Classic, and the Salesforce app.

These global variables return a string that uniquely identifies the current user interface context. The possible values for `$User.UITheme` and `$User.UIThemeDisplayed` are the same:

- `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`—Salesforce mobile app theme
- `Theme4u`—Lightning Console theme
- `PortalDefault`—Salesforce Customer Portal theme
- `Webstore`—Salesforce AppExchange theme

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, `$User.UIThemeDisplayed` returns a different value. In general, your code should use `$User.UIThemeDisplayed`.

The simplest way to use these theme globals is to use one in a Boolean expression, like `{! $User.UIThemeDisplayed == "Theme3" }`, in the `rendered` attribute of a component. The component will only display if the page appears in the desired user interface context.

```
<apex:outputText value="This is Salesforce Classic."
    rendered="{! $User.UIThemeDisplayed == 'Theme3' }"/>
```

Although you can use this technique on individual user interface elements, it’s usually more efficient if you wrap larger chunks of markup into an `<apex:outputPanel>` or similar block-level component, and then create separate blocks for each different UI you want to present. Then place the theme test on the `rendered` attribute of the blocks, rather than the individual components. Not only should this perform better, your code will be less complicated.

```
<apex:outputPanel rendered="{! $User.UIThemeDisplayed == 'Theme3' }">

    <apex:outputText value="This is Salesforce Classic."/>
    <apex:outputText value="These are multiple components wrapped by
an outputPanel."/>
</apex:outputPanel>
<apex:outputPanel rendered="{! $User.UIThemeDisplayed == 'Theme4d' }">

    <apex:outputText value="Everything is simpler in Lightning
Experience."/>
</apex:outputPanel>
```

Another strategy you can use this with is to dynamically select a stylesheet to include on your page, and provide a different stylesheet for each theme. This is a bit trickier than you might think, because the `<apex:stylesheet>` tag doesn't have a `rendered` attribute of its own. In this case, you must wrap the stylesheet components within another component that does have a `rendered` attribute. Here's an example of how to provide a different stylesheet for each of the three modern themes supported by Salesforce.

```
<apex:page standardController="Account">

    <!-- Salesforce Classic "Aloha" theme -->
    <apex:variable var="uiTheme" value="classic2010Theme"
        rendered="{!$User.UIThemeDisplayed == 'Theme3'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
                                         'classic-styling.css')}" />
    </apex:variable>

    <!-- Lightning Desktop theme -->
    <apex:variable var="uiTheme" value="lightningDesktop"
        rendered="{!$User.UIThemeDisplayed == 'Theme4d'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
                                         'lightning-styling.css')}"
/>
    </apex:variable>

    <!-- Salesforce mobile theme -->
    <apex:variable var="uiTheme" value="SalesforceApp"
        rendered="{!$User.UIThemeDisplayed == 'Theme4t'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
                                         'mobile-styling.css')}" />
    </apex:variable>

    <!-- Rest of your page -->

    <p>
        Value of $User.UIThemeDisplayed: {! $User.UIThemeDisplayed }
    </p>
</apex:page>
```

Beyond the Basics

This is an unusual way to use `<apex:variable>`, because we're not actually interested in the value of the variable created. Instead we just want a component that doesn't render any output of its own to wrap the `<apex:stylesheet>` component. You can think of this as `<apex:variable>` "lending" its `rendered` attribute to the wrapped `<apex:stylesheet>` component.

It's a good thing we don't care about the variable itself, because another unusual aspect of wrapping the `<apex:variable>` component around something else is that the variable isn't actually created! Feature or bug? Let's call it...undefined behavior, and avoid using the `uiTheme` variable elsewhere.

Detecting and Responding to the User Experience Context in JavaScript

Detecting the current user experience context in JavaScript code is important if you're using JavaScript heavily in your pages and apps. It's especially important for using the right technique to manage navigation in your JavaScript code.

To detect what the user sees in JavaScript, we use a similar method to determining the current user experience context in Visualforce. Call the `UITheme.getUITheme()` global variable to return a value that identifies the current user interface theme.

Here the code checks if the current user experience context is the Lightning Experience theme.

```
function isLightningDesktop() {
  return UITheme.getUITheme === "Theme4d";
}
```

Determining the User Experience Context in Apex

Use the `UserInfo.getUiTheme()` and `UserInfo.getUiThemeDisplayed()` system methods to determine the current user experience context in Apex code. You can use them when your controller action methods or properties need to behave differently in different contexts.

The following example illustrates how to use these methods by making them available via getter methods in a controller extension.

```
public with sharing class ForceUIExtension {

    // Empty constructor, required for Visualforce controller extension

    public ForceUIExtension(ApexPages.StandardController controller)
{ }

    // Simple accessors for the System.UserInfo theme methods
    public String getContextUserUiTheme() {
        return UserInfo.getUiTheme();
    }
    public String getContextUserUiThemeDisplayed() {
        return UserInfo.getUiThemeDisplayed();
    }

}
```

You could of course work with the values in your Apex code, rather than directly returning the method call results.

These Apex system methods return a string that uniquely identifies the current user interface context. The possible values returned by these methods are the same as those returned by the `$User.UITheme` and `$User.UIThemeDisplayed` global variables.

- `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`—Salesforce mobile app theme
- `Theme4u`—Lightning Console theme
- `PortalDefault`—Salesforce Customer Portal theme
- `Webstore`—Salesforce AppExchange theme

Using these methods in server-side controller code should be rare, at least compared to providing different Visualforce markup or JavaScript code. It's a best practice for your controller and controller extension code to be neutral in terms of the UX context. Let your front end code, whether Visualforce or JavaScript, handle the user interface differences.

Querying for Lightning Experience via SOQL and API Access

Although we don't recommend this technique, you can query for the current user's preferred user experience directly using SOQL.

The basic SOQL query is the following.

```
SELECT UserPreferencesLightningExperiencePreferred FROM User WHERE Id
 = 'CurrentUserId'
```

The result is a raw preference value, which you need to convert into something useable.

Here's just about the simplest possible Visualforce page that runs the above SOQL query and displays the result on the page.

```
<apex:page>

<script src="/soap/ajax/36.0/connection.js"
type="text/javascript"></script>
<script type="text/javascript">

    // Query for the preference value
    sforce.connection.sessionId = '{! $Api.Session_ID }';
    var uiPrefQuery = "SELECT Id,
UserPreferencesLightningExperiencePreferred " +
                      "FROM User WHERE Id = '{! $User.Id }'";
    var userThemePreferenceResult =
sforce.connection.query(uiPrefQuery);

    // Display the returned result on the page
    document.addEventListener('DOMContentLoaded', function(event){
       document.getElementById('userThemePreferenceResult').innerHTML
 =
            userThemePreferenceResult;
    });
</script>
```

```
<h1>userThemePreferenceResult (JSON)</h1>

<pre><span id="userThemePreferenceResult"/></pre>

</apex:page>
```

Querying for the user's Lightning Experience preference directly is discouraged. The result tells you what the user's current preference *setting* is, not what user experience actually is on their screen. There are several use cases where the preference value might not reflect the user experience that's actually being delivered. To determine the actual user experience being delivered in the current request, use `$User.UIThemeDisplayed` or `UserInfo.getUiThemeDisplayed()`.

CHAPTER 38 Managing Navigation

In this chapter ...

App flow and navigation is in many ways the heart of application design. Visualforce provides a number of ways to add navigation elements and to direct application flow. Lightning Experience adds its own application flow, navigation elements, and mechanisms for affecting where users go as they use Salesforce.

The good news is that “classic” Visualforce navigation continues to work. The better news is that your Visualforce pages can take advantage of the new Lightning Experience mechanisms, too.

Navigation in Lightning Experience

Before we talk about the details of Visualforce navigation, and how you create it so that it works in Salesforce Classic and Lightning Experience, let's talk a little about navigation in general. What do we actually mean by "navigation"?

The first thing we might mean by navigation is user interface elements on the screen. You click something, and something happens. For example, you click the Accounts item in the navigation menu, and you go to the Accounts object home page. You click the New button, and a record entry form appears. You choose a custom action from a quick actions menu, and you launch a custom process. And so on. Those buttons and menu items are navigation elements.

The design of the navigation system, the user interface in Lightning Experience, is very different from Salesforce Classic. We're not going to talk about those differences here, but you'll want to be familiar with where everything moves when you switch between the two user experiences. You can learn more about that right here in Trailhead, in the Navigating Lightning Experience and Setup unit.

Another, less visible kind of navigation is the "something happens" part of the above. Behind the scenes, Salesforce decides what's going to happen when you select an item in a menu, or click a link or button. Much of this navigation is built into Salesforce already, while other aspects are customizable—for example, overriding actions with Visualforce pages. But all of this navigation is managed by code written by Salesforce.

And then there's navigation in your own apps—apps that use *your* code to control application flow. When your custom action opens a form and the user clicks save, where do you go? When your running code makes a decision about where the user should go next, and sends them there. *This* is what we're going to talk about in this unit.

Classic Visualforce Navigation

"Classic" Visualforce navigation can be boiled down to "what happens at the end of an action method." Action methods return a `PageReference` object with the details of where the user is to be navigated to, and then the Visualforce framework handles the details of sending the right response back to the user's browser. And, great news, all of this still works.

Remember also that the Standard Controller returns a `PageReference` from its action methods. So, your existing navigation, whether you're using the Standard Controller or your own custom controller code, continues to work as you expect.

Modern Visualforce Navigation

So, if classic Visualforce navigation works, why are we still talking about this? What are we even having a conversation about? We just want to say one word to you. Just one word. Are you listening? . . . "JavaScript."

There's a great future in JavaScript—and that future is here today. Many Visualforce developers are using JavaScript heavily in their apps, and that use continues to grow. Classic Visualforce works, and will continue to work for a long time. But as developers adopt Visualforce features such as Remote Objects and JavaScript remoting, more of their apps' behavior migrates from the server side, where `PageReference` is the rule, to the browser and JavaScript, where there's no such thing as a `PageReference`.

In the Lightning Experience (and the Salesforce mobile app) world, there are rules and tools for building navigation in JavaScript. We'll cover the rules, which are mostly about what not to do, in a little bit. Let's talk about the right way to do things first.

Lightning Experience manages navigation using events. The navigation event framework is made available as a JavaScript utility object that provides a number of functions that make creating programmatic navigation straightforward. The `sforce.one` object is automatically added to Visualforce pages when they run in Lightning Experience. This object provides a number of functions that trigger navigation events when the functions are called. To use these functions, you can call them directly from your page's JavaScript code, or you can attach calls as click (or other) handlers to elements on the page.

Important: The `sforce.one` object isn't available in Salesforce Classic. Any code that uses it should test for the existence of `sforce.one` first.

The `sforce.one` object provides the following functions. Reference the function using dotted notation from the `sforce.one` object. For example: `sforce.one.navigateToSObject(...)`.

Function	Description
`back([refresh])`	Navigates to the previous state that's saved in the `sforce.one` history. It's equivalent to clicking a browser's Back button.
`navigateToSObject(recordId[, view])`	Navigates to an sObject record, specified by `recordId`.
`navigateToURL(url[, isredirect])`	Navigates to the specified URL.
`navigateToFeed(subjectId, type)`	Navigates to the feed of the specified `type`, scoped to the `subjectId`.
`navigateToFeedItemDetail(feedItemId)`	Navigates to the specific feed item, `feedItemId`, and any associated comments.

Function	Description
`navigateToRelatedList( relatedListId, parentRecordId)`	Navigates to a related list for the `parentRecordId`.
`navigateToList( listViewId, listViewName, scope)`	Navigates to the list view that's specified by the `listViewId`, which is the ID of the list view to be displayed.
`createRecord( entityName[, recordTypeId])`	Opens the page to create a new record for the specified `entityName`, for example, "Account" or "MyObject__c".
`editRecord(recordId)`	Opens the page to edit the record specified by `recordId`.

For additional details about using these functions, and the parameters they accept, see Navigation and Messaging with the `sforce.one` Object in this unit's Resources.

Navigation Gotchas, and How to Fix Them

The first rule for building Visualforce navigation in JavaScript is: do not set `window.location` directly. The second rule for building Visualforce navigation in JavaScript is: *do not* set `window.location` directly.

Don't Set **`window.location`** Directly

OK, gratuitous repetition and movie reference aside, what's the big deal here? It's pretty simple. When in Lightning Experience your page doesn't *have* a `window.location` to set! Remember all the earlier discussion about the Visualforce "container," and being in an iframe, and Lightning Experience being some kind of health club? (SPA—single-page application.) This is one of the things that falls out of it. The Visualforce iframe doesn't have direct access to the `window.location` value, so you can't set it. If your code depends on setting it, it'll break. That is, actions that fire navigation by setting `window.location` will simply stop navigating to wherever you were expecting to go.

There's actually a way around this restriction, but you shouldn't use it. The reason is if you bypass the navigation functions in `sforce.one`, your navigation events won't be tracked in the Lightning Experience navigation stack. That stack provides useful features, like Back buttons that account for redirects and the like. A number of features in Lightning Experience (and especially in the Salesforce mobile app) depend on that stack containing all navigation events. It's worth making sure you use it correctly.

The Salesforce Classic Issue

So, yeah, there's this one…thing. Unfortunately, the `sforce.one` utility object isn't available when your page runs in Salesforce Classic. In that context, you *have* to use `window.location`. The good news is, in Salesforce Classic, `window.location` is available. The bad news is, this limitation means you'll have to add an ugly `if` block to your code. Consider wrapping your navigation functions in utility methods that deal with this complexity, so that your main navigation logic can be straightforward.

Static URLs

Don't use static URLs to Salesforce resources. That is, if you're adding a link to edit a Contact record, don't create the link by building a string with a static pattern like `link = '/' + accountId + '/e'`. In some contexts this works, but in others it doesn't. Instead, try one of these approaches:

- In Visualforce markup, use `{!URLFOR($Action.Contact.Edit, recordId)}`
- In JavaScript, use `navigateToSObject(recordId)`

There are actions and functions for viewing, creating, editing, and so on. Use them, rather than URL strings.

CHAPTER 39 Understanding Important Visual Design Considerations

In this chapter ...

Visualforce pages look the same whether they are running in Salesforce Classic or Lightning Experience, unless you rework them to adapt to the appropriate user interface context. Built-in Visualforce components that display user interface elements aren't easily restyled to match the Lightning Experience look-and-feel.

Specifically, the HTML that's rendered by the built-in Visualforce components doesn't change when the page displays in Lightning Experience, and the Salesforce Classic stylesheets those components use is, by default, loaded by the page. The effect is that pages that use `<apex:inputField>`, `<apex:outputField>`, the `<apex:pageBlock>` components, and other coarse- and fine-grained components that match the Salesforce Classic visual design, still match that visual design. You get a little slice of Salesforce Classic in the middle of your Lightning Experience.

It's our general recommendation that—for now, for existing pages—you don't try to adapt them to match the visual design of Lightning Experience. Lightning Experience is still evolving, and matching its styling yourself means you're chasing a moving target. That's work.

Nevertheless, in some cases you'll want some pages to match more closely with Lightning Experience visuals. For new pages, or if you're willing to do some work, there are some great tools for creating pages that fit in perfectly with Lightning Experience.

Affecting the Styling of Standard Components

Visualforce provides a range of options for adjusting or overriding the styling of the standard components. If your goal is to make modest changes to the appearance of these components, the effort to use these options is similarly modest. Let's look at a few of the tools you have available for affecting styling.

Styling Individual Components

Visualforce components that produce HTML have pass-through `style` and `styleClass` attributes. These attributes allow you to use your own styles and style classes to control the look and feel of the resulting HTML. `style` allows you to set styles directly on a component, while `styleClass` lets you attach classes for styles defined elsewhere. For example, the following code sets the class of the `<apex:outputText>` and applies a style.

```
<apex:page>

    <style type="text/css">
        .asideText { font-style: italic; }
    </style>

    <apex:outputText style="font-weight: bold;"
        value="This text is styled directly."/>

    <apex:outputText styleClass="asideText"
        value="This text is styled via a stylesheet class."/>

</apex:page>
```

Adding a Custom Stylesheet

You can add your own custom stylesheets to any Visualforce page using static resources and the `<apex:stylesheet>` tag. For example, to add a stylesheet that's been uploaded as a static resource named "AppStylesheet", add the following to your page.

```
<apex:stylesheet value="{!$Resource.AppStylesheet}"/>
```

You can then refer to any of the styles contained in the stylesheet, and reference them in Visualforce tag `styleClass` attributes, as we did with the `asideText` style previously.

This is the recommended method for adding CSS style definitions to Visualforce pages, because it shares the stylesheet between pages, and minimizes the markup you need to add to each page.

The exception to this method for adding a stylesheet is the Salesforce Lightning Design System. The Lightning Design System is a fantastic all-new toolkit for styling your pages, and we'll talk about it in detail shortly.

Although you can upload the Lightning Design System as a static resource and reference it using `<apex:stylesheet>`, there's an easier way: Just include `<apex:slds />` anywhere in the markup of your page.

Different Styles in Lightning Experience

To load a custom stylesheet only when your page is running in Lightning Experience, use the following markup. This is similar to the Visualforce markup example in Sharing Visualforce Pages Between Classic and Lightning Experience.

```
<apex:page standardController="Account">

    <!-- Base styles -->
    <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
'app-styles.css')}" />

    <!-- Lightning Desktop extra styles -->
    <apex:variable var="uiTheme" value="lightningDesktop"
        rendered="{!$User.UIThemeDisplayed == 'Theme4d'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
'lightning-styling.css')}" />
    </apex:variable>

    <!-- Rest of your page -->

</apex:page>
```

OK, these are tools. Let's look at a few techniques for using them next.

Styling Strategies and Recommendations

To create Visualforce pages that match the Lightning Experience visual design, create new pages using the Lightning Design System. There are a couple of ways to use the Lightning Design System in your Visualforce pages.

Before we get to specifics, let's think at a higher level and consider the different strategies for applying Lightning Experience styling to your pages. In particular, let's talk about your existing pages.

There are two ways to affect the styling of existing pages to make them look more like Lightning Experience.

- Change the *markup* to apply new styling—make changes in your pages.
- Change the *styling rules* for existing markup—make changes in your stylesheets.

These aren't either / or. You can use them individually or in combination.

Correctly using the Lightning Design System means using the Lightning Design System stylesheets with all-new markup for your Visualforce pages. Again, this is the only *supported* method for matching the Lightning Experience visual design.

To do this, you can either download the Lightning Design System stylesheets from their website and use them as you would any other stylesheet, or you can add the `<apex:slds>` component to the markup of your Visualforce page. The `<apex:slds>` component allows you to reference Lightning Design System stylesheets without uploading them as a static resource, simplifying the syntax of your page and preventing you from hitting the 250-mb static resource limit.

Using `<apex:slds>` comes with its own set of guidelines and considerations. If you want to know more, work your way through the Lightning Design System badge or check out the link to the *Visualforce Developer Guide* in the Resources section.

It's also possible to add the Lightning Design System stylesheets, and revise your pages to use them. How much work this is depends on how closely you want to match Lightning Experience as well as the specific markup and components in your code. While it's possible to achieve decent results this way, however, it's not an approach we recommend. The Lightning Design System was designed to be applied to specific markup, and that's simply not what Visualforce emits. There's an "impedance mismatch" that, while not fatal, is definitely a serious rock in your shoe when you take this path.

Finally, there's the other approach: adding new rules and styles to your existing (or a new) stylesheet to make your existing markup look more like Lightning Experience. If your page is already mostly styled with your own stylesheets, this approach might work well for you. If instead you're mostly using the built-in Visualforce components and the Salesforce Classic styling, it requires you to override the styles from the Salesforce Classic stylesheet.

While this is technically possible, we want to discourage you from taking this approach. It introduces dependencies into your markup and styles that you don't want to have. These dependencies are on the structure, IDs, and classes of the HTML rendered by the built-in Visualforce components. We want to be really clear here: the HTML rendered by the built-in Visualforce components is an internal implementation detail, subject to change without notice. If you have dependencies on it in your own stylesheets, your styling *will* eventually break.

The Salesforce Lightning Design System

The Lightning Design System is a design framework for building enterprise apps that look like Lightning Experience. It includes a sophisticated CSS framework, a collection of graphic assets, and the Salesforce Sans font. You can use the Lightning Design System to build pages and apps that look gorgeous and perfectly match the Lightning Experience user interface.

The Lightning Design System was designed to make it easy for customers and partners to match the Lightning Experience look and feel. It also includes tools that make it possible to customize the look and feel to match your own brand—colors and so on—while still remaining consistent with overall Lightning Experience design.

The Lightning Design System is so big and so exciting that. . . we're not going to go into the details of using the Lightning Design System here. Because we've written a whole module about using it, Lightning Design System. It explains how to get the Lightning Design System, the basic concepts of using it to design pages, and how to apply those concepts to building Lightning Experience apps with Visualforce.

Lightning Design System is a big module—you'll have to work to earn that badge. While we do want to save the details for that module, we don't want to leave you totally hanging here. So, let's cover how you use the Lightning Design System with Visualforce in a general way.

The first thing to know is that the Lightning Design System assumes a new markup structure and styling classes. For this reason it's best used with new pages and apps. It's built around the capabilities of modern browsers, and takes advantage of the latest best practices for markup and styling. As much as we all love it, Visualforce has been around a while. Between the HTML it generates and static code in customer pages, most organizations will find it challenging to apply the Lightning Design System to existing pages.

The Lightning Design System module is focused on creating new pages and apps, and scoring that badge is the best way to learn about it. After finishing that module you'll have an understanding of both how to use Lightning Design System and how to plan development around it.

CHAPTER 40 Knowing Which Features to Avoid in Lightning Experience

In this chapter ...

There are a limited number of Visualforce components that we recommend you avoid on pages used in Lightning Experience. Additionally, a few features of Visualforce behave differently when used in Lightning Experience. Finally, there are a few places in Lightning Experience where you can't use Visualforce pages or apps, or where they might not function as expected.

Lightning Experience is still evolving and growing, and—Safe Harbor alert!—we hope to shrink this list as time goes on.

Lightning Experience Header and Navigation Menu Can't Be Suppressed

Visualforce pages always display with the standard Lightning Experience user interface when they run in Lightning Experience. There's no way to suppress or alter the Lightning Experience header or sidebar. In particular, the `showHeader` and `sidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience.

This behavior is intentional. Apps that display in Lightning Experience are *Lightning Experience* apps. If you need to provide a completely custom interface for your app, you'll need to run it in Salesforce Classic.

Salesforce Classic Header and Sidebar Are Always Suppressed

The standard Salesforce Classic header and sidebar are always suppressed for pages when they're displayed in Lightning Experience. In particular, the `showHeader` and `sidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience.

Pages behave as though the `showHeader` and `sidebar` attributes of `<apex:page>` are both set to `false`.

Note: The `standardStylesheets` attribute of `<apex:page>`, which determines whether to include or suppress the standard Salesforce Classic stylesheets, is unaffected by Lightning Experience. That is, it defaults to `true` in Lightning Experience, but you're able to change it.

`<apex:relatedList>` and Blacklisted Related Lists

There are a number of related lists that aren't supported in Lightning Experience. These related lists are "blacklisted," which means they are explicitly prevented from being used. As you might expect, these same related lists are blacklisted in Visualforce with the `<apex:relatedList>` tag.

See "Data Access and Views: What's Different or Not Available in Lightning Experience" in the online help for details of which related lists aren't supported in Lightning Experience.

Avoid `<apex:iframe>`

While it's not impossible to use `<apex:iframe>` on a Visualforce page in Lightning Experience, we recommend avoiding it.

Visualforce pages are wrapped in their own iframe when displayed in Lightning Experience. As discussed at length in Exploring the Visualforce App Container, this has a number of significant implications for how the page behaves. Adding an additional level to the iframe stack increases the complexity of the environment.

If you really understand iframes and how they affect the DOM and JavaScript, you can manage this complexity. But unless you've already been working with nested iframes, it's more likely that you'll have difficult to debug problems. For this reason, we suggest you avoid this tag on pages that are used in Lightning Experience.

No, Really, Don't Set `window.location` Directly

We probably sound like a broken record at this point, but it's important. If your page's JavaScript code is setting the `window.location` variable directly, that won't work when the page is displayed in Lightning Experience. You *must* modify this code for the page to work in Lightning Experience.

See the Managing Navigation unit for details.

`sforce.one` Isn't Salesforce Mobile-Only

The `sforce.one` JavaScript utility object is available to Visualforce pages in both the Salesforce mobile app and Lightning Experience. If you've been using the presence of the `sforce.one` object as a way to tell if your page is running in a mobile or desktop context, you need to update your code.

Use one of the documented methods to distinguish between the Salesforce Classic, the Salesforce mobile app, and Lightning Experience environments. Supported techniques are available in Visualforce, Apex, and JavaScript.

For the full details, see the Sharing Visualforce Pages Between Classic and Lightning Experience unit.

Changes With Action Overrides

Perhaps the most significant change, in terms of things that might be hard to work around, Visualforce overrides of standard actions are slightly different in Lightning Experience compared to Salesforce Classic. Any override for the object list action won't be accessible in Lightning Experience.

Specifically, there are six standard actions you can override for most standard and all custom objects in Salesforce Classic:

- Object tab
- Object list
- Record view
- Record edit
- Record create
- Record delete

In Lightning Experience the first two actions have been combined into one page, object home. Object home is similar to object list, along with some elements of object tab, like recent items. Other elements, such as reports or tools, have moved to other parts of the user interface.

Regardless of the user interface settings in your organization, both object tab and object list are available to be overridden in Setup. Overriding the object tab action overrides the object home page in Lightning Experience, as expected.

However, when in Lightning Experience, the object list action isn't accessible in the user interface, so there's no way to fire it. If your organization has overridden the object list action for any object, that functionality won't be available when users are using Lightning Experience. If there are essential features in that override, you'll need to find another way to make them available.

This table lists the standard actions you can override for an object in Setup, and the action that's overridden in the three different user experiences.

Override in Setup	Salesforce Classic	Lightning Experience	Salesforce App
Tab	object tab	object home	search
List	object list	**n/a**	object home
View	record view	record home	record home
Edit	record edit	record edit	record edit
New	record create	record create	record create

Override in Setup	Salesforce Classic	Lightning Experience	Salesforce App
Delete	record delete	record delete	record delete

Note: "n/a" doesn't mean you can't *access* the standard behavior, and it doesn't mean you can't *override* the standard behavior. It means you can't *access the override*. It's the override's functionality that's not available.

INDEX