

salesforce

---

# Single Sign-On Implementation Guide

Salesforce, Spring '18



 @salesforcedocs

Last updated: March 23, 2018



# CONTENTS

Single Sign-On	1
SAML	2
Configure SAML Settings for Single Sign-On	3
Working With Your Identity Provider	7
Identity Provider Values	8
Customize SAML Start, Error, Login, and Logout Pages	12
Example SAML Assertions	13
View and Edit Single Sign-On Settings	24
Validating SAML Settings for Single Sign-On	25
SAML Assertion Validation Errors	26
Reviewing the SAML Login History	28
About Just-in-Time Provisioning for SAML	29
Just-in-Time Provisioning Requirements and SAML Assertion Fields	29
Just-in-Time Provisioning and SAML Assertion Fields for Portals	31
Just-in-Time Provisioning for Communities	34
Just-in-Time Provisioning Errors	38
Configure SSO Across Multiple Salesforce Orgs	41
Best Practices and Tips for Implementing Single Sign-On	44
Configuring SSO for Mobile and Desktop Apps Using SAML and OAuth	47
Configuring SAML SSO for a Canvas App	52
Enable Single Sign-On for Portals	60
Delegated Authentication Single Sign-On	61
Configure Salesforce for Delegated Authentication	62
Sample Delegated Authentication Implementations	64
Single Logout	65
Configure SAML Settings for Single Logout Where Salesforce Is the Service Provider	66
Configure SAML Settings for Single Logout Where Salesforce Is the Identity Provider	69
Configure OpenID Connect Settings for Single Logout Where Salesforce Is the Relying Party	71
Configure OpenID Connect Settings for Single Logout Where Salesforce Is the OpenID Connect Provider	73
Frequently Asked Questions	75

**Contents**

Index ..... 77

# SINGLE SIGN-ON

Single sign-on (SSO) lets users access authorized network resources with one login. You validate usernames and passwords against your corporate user database or other client app rather than Salesforce managing separate passwords for each resource.

Salesforce offers the following ways to use SSO.

- Federated authentication using Security Assertion Markup Language (SAML) lets you send authentication and authorization data between affiliated but unrelated web services. You can log in to Salesforce from a client app. Salesforce enables federated authentication for your org automatically.
- Delegated authentication SSO integrates Salesforce with an authentication method that you choose. You can integrate authentication with your LDAP (Lightweight Directory Access Protocol) server or use a token instead of a password for authentication. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Delegated authentication offers the following benefits.

- Uses a stronger form of user authentication, such as integration with a secure identity provider
- Makes your login page private and accessible only behind a corporate firewall
- Differentiates your org from all other companies that use Salesforce to reduce phishing attacks

You must contact Salesforce to enable delegated authentication before you can configure it on your org.

- Authentication providers let your users log in to your Salesforce org using their login credentials from an external service provider. Salesforce supports the OpenID Connect protocol, which lets users log in from any OpenID Connect provider, such as Google, PayPal, and LinkedIn. When an authentication provider is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

When you have an external identity provider and configure SSO for your Salesforce org, Salesforce is then acting as a service provider. You can also enable Salesforce as an [identity provider](#) and use SSO to connect to a different service provider. Only the service provider needs to configure SSO.

The Single Sign-On Settings page displays which version of SSO is available for your org. To learn more about SSO settings, see [Configure SAML Settings for Single Sign-On](#). For more information about SAML and Salesforce security, see the [Security Implementation Guide](#).

## Benefits of SSO

Implementing SSO brings several advantages to your org.

- **Reduced administrative costs**—With SSO, users memorize a single password to access network resources and external apps and Salesforce. When accessing Salesforce from inside the corporate network, users log in seamlessly and aren't prompted for a username

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data

or password. When accessing Salesforce from outside the corporate network, the users' corporate network login works to log them in. With fewer passwords to manage, system admins receive fewer requests to reset forgotten passwords.

- **Leverage existing investment**—Many companies use a central LDAP database to manage user identities. You can delegate Salesforce authentication to this system. Then when users are removed from the LDAP system, they can no longer access Salesforce. Users who leave the company automatically lose access to company data after their departure.
- **Time savings**—On average, users take 5–20 seconds to log in to an online app. It can take longer if they mistype their username or password and are prompted to reenter them. With SSO in place, manually logging in to Salesforce is avoided. These saved seconds reduce frustration and add up to increased productivity.
- **Increased user adoption**—Due to the convenience of not having to log in, users are more likely to use Salesforce regularly. For example, users can send email messages that contain links to information in Salesforce, such as records and reports. When the recipient of the email message clicks the links, the corresponding Salesforce page opens.
- **Increased security**—All password policies that you've established for your corporate network are in effect for Salesforce. Sending an authentication credential that's only valid for a single time also increases security for users who have access to sensitive data.

## SAML

Salesforce Identity uses the XML-based Security Assertion Markup Language (SAML) protocol for single sign-on into Salesforce from a corporate portal or identity provider. With SAML, you can transfer user information between services, such as from Salesforce to Microsoft 365.

The identity provider performs most of the work to set up single sign-on (SSO).

1. Establish a SAML identity provider and [gather information](#) about how they connect to Salesforce. The identity provider sends SSO requests to Salesforce.
2. Provide information to your identity provider, such as the [URLs for the start and logout pages](#).
3. Configure Salesforce using the instructions in [Configure SAML Settings for Single Sign-On](#). Only this step takes place in Salesforce.

Your identity provider sends SAML assertions to Salesforce using the SAML Web Single Sign-on Browser POST profile. Salesforce sends SAML responses to the identity provider login URL specified under Setup by entering *Single Sign-On* in the **Quick Find** box, then selecting **Single Sign-On Settings**. Salesforce receives the assertion, verifies it against your Salesforce configuration, and, if the assertion is true, allows SSO.

If you have problems with the SAML assertion after you configure Salesforce for SAML, use the SAML Assertion Validator to [validate the SAML assertion](#). You can obtain a SAML assertion from your identity provider.

If your users can't log in using SAML, [review the SAML login history](#) to determine why. Sharing the login history with your identity provider helps resolve problems quickly.

Click **Download Metadata** to download an XML file of your SAML configuration settings to send to your identity provider. The identity provider can then upload these configuration settings to connect to your Salesforce orgcommunity.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application
- AND
- Modify All Data

# CONFIGURE SAML SETTINGS FOR SINGLE SIGN-ON

From this page, you can configure your org to use single sign-on. You can also set up just-in-time provisioning. Work with your identity provider to properly configure these settings. For more information about single sign-on, see [Single Sign-On](#). For more information about just-in-time provisioning, see [About Just-In-Time Provisioning](#).

To configure SAML settings for single sign-on from your corporate identity provider to Salesforce:

1. [Gather information from your identity provider.](#)
2. [Provide information to your identity provider.](#)
3. [Set up single sign-on.](#)
4. [Set up an identity provider to encrypt SAML assertions \(optional\).](#)
5. [Enable Just-in-Time user provisioning \(optional\).](#)
6. [Edit the SAML JIT handler](#) if you selected `Custom SAML JIT with Apex Handler` for Just-in-Time provisioning.
7. [Test the single sign-on connection.](#)

## Set up single sign-on

1. In Salesforce, from Setup, enter *Single Sign-On Settings* in the `Quick Find` box, then select **Single Sign-On Settings**, and click **Edit**.
2. Select `SAML Enabled`. You must enable SAML to view the SAML single sign-on settings.
3. Specify the SAML version used by your identity provider.
4. Click **Save**.
5. In SAML Single Sign-On Settings, click the appropriate button to create a configuration, as follows.
  - **New** - Specify all settings manually.
  - **New from Metadata File** - Import SAML 2.0 settings from an XML file from your identity provider. This option reads the XML file and uses it to complete as many of the settings as possible.
    - 📌 **Note:** If your XML file contains information for more than one configuration, the first configuration that occurs in the XML file is used.
  - **New from Metadata URL** - Import SAML 2.0 settings from a public URL. This option reads the XML file at a public URL and uses it to complete as many of the settings as possible. The URL must be added to Remote Site Settings to access it from your Salesforce org.
6. Give this setting a **Name** for reference within your org.  
Salesforce inserts the corresponding **API Name** value, which you can customize if necessary.
7. Enter the `Issuer`. Often referred to as the entity ID for the identity provider.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

### USER PERMISSIONS



To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application AND Modify All Data

## Configure SAML Settings for Single Sign-On

8. If your Salesforce org has [domains](#) deployed, specify whether you want to use the base domain (<https://saml.salesforce.com>) or the custom domain for the **Entity ID**. You must share this information with your identity provider.
  -  **Tip:** Generally, use the custom domain as the entity ID. If you already have single sign-on configured before deploying a domain, the base domain is the entity ID. If you are providing Salesforce to Salesforce services, you must specify the custom domain.
9. For the **Identity Provider Certificate**, use the **Browse** button to locate and upload the authentication certificate issued by your identity provider. The certificate size can't exceed 4 KB. If it does, try using a DER encoded file to reduce the size.
10. For the **Request Signing Certificate**, select the certificate you want from the ones saved in your **Certificate and Key Management** settings.
11. For the **Request Signature Method**, select the hashing algorithm for encrypted requests, either **RSA-SHA1** or **RSA-SHA256**.
12. Optionally, if the identity provider encrypts SAML assertions, select the **Assertion Decryption Certificate** they're using from the ones saved in your **Certificate and Key Management** settings. This field is available only if your org supports multiple single sign-on configurations. For more information, see [Set up an identity provider to encrypt SAML assertions](#).
13. For the **SAML Identity Type**, **SAML Identity Location**, and other fields described in [Identity Provider Values](#), specify the values provided by your identity provider as appropriate.
14. For the **Service Provider Initiated Request Binding**, select the appropriate value based on the information provided by your identity provider.
15. For SAML 2.0, if your identity provider has specific login or logout 2.0 pages, specify them in **Identity Provider Login URL** and **Identity Provider Logout URL**, respectively.
  -  **Note:** These fields appear in Developer Edition and sandbox organizations by default and in production organizations only if My Domain is enabled. The fields do not appear in trial organizations or sandboxes linked to trial organizations.
16. For the **Custom Error URL**, specify the URL of the page that the users are directed to if there's an error during SAML login. It must be a publicly accessible page, such as a public site Visualforce page. The URL can be absolute or relative.
17. Optionally, set up Just-in-Time user provisioning. For more information, see [Enable Just-in-Time user provisioning](#) and [About Just-in-Time Provisioning for SAML](#).
18. Click **Save**.


Click **Download Metadata** to download an XML file of your SAML configuration settings to send to your identity provider. The identity provider can then upload these configuration settings to connect to your Salesforce orgcommunity.

## Set up an identity provider to encrypt SAML assertions

---

When Salesforce is the service provider for inbound SAML assertions, you can pick a saved certificate to decrypt inbound assertions from third party identity providers. You need to provide a copy of this certificate to the identity provider.

1. In the Single Sign-On Settings page in Setup, add a new SAML configuration.
2. In the **Assertion Decryption Certificate** field, specify the certificate for encryption from the ones saved in your **Certificate and Key Management** settings.

 **Note:** If you don't see the **Assertion Decryption Certificate** field you need to enable multiple single sign-on for your organization. (Applies to orgs created before the Summer '13 release that aren't using SAML 1.1). To enable multiple



## Configure SAML Settings for Single Sign-On

single sign-on configurations, select **Enable Multiple Configs** on the **Single Sign-On Settings** page. If this setting has already been enabled, the field appears, and you won't see the **Enable Multiple Configs** button.

3. Set the `SAML Identity Location` to the element where your identifier is located.
4. When you save the new SAML configuration, your org's SAML settings value for the `Salesforce Login URL` (also known as the "Salesforce ACS URL") changes. Get the new value (from the Single Sign-On Settings page in Setup), and click the name of the new SAML configuration. The value is in the `Salesforce Login URL` field.
5. The identity provider must use the `Salesforce Login URL` value.
6. You also need to provide the identity provider with a copy of the certificate selected in the `Assertion Decryption Certificate` field to use for encrypting assertions.

## Enable Just-in-Time user provisioning

---

1. In SAML Single Sign-On Settings, select `User Provisioning Enabled`.
  - `Standard` - This option allows you to provision users automatically using attributes in the assertion.
  - `Custom SAML JIT with Apex handler` - This option provisions users based on logic in an Apex class.
2. If you selected `Standard`, click **Save** and [test the single sign-on connection](#). If you selected `Custom SAML JIT with Apex handler`, proceed to the next step.
3. In the `SAML JIT Handler` field, select an existing Apex class as the SAML JIT handler class. This class must implement the [SamJitHandler interface](#). If you do not have an Apex class, you can generate one by clicking `Automatically create a SAML JIT handler template`. You must edit this class and modify the default content before using it. For more information, see [Edit the SAML JIT handler](#).
4. In the `Execute Handler As` field, select the user that runs the Apex class. The user must have "Manage Users" permission.
5. Just-in-time provisioning requires a Federation ID in the user type. In `SAML Identity Type`, select `Assertion contains the Federation ID from the User object`. If your identity provider previously used the Salesforce username, communicate to them that they must use the Federation ID.
6. Click **Save**.

## Edit the SAML JIT handler


---

1. From Setup, enter `Apex Classes` in the `Quick Find` box, then select **Apex Classes**.
2. Edit the generated Apex SAML JIT handler to map fields between SAML and Salesforce. In addition, you can modify the generated code to support the following:
  - Custom fields
  - Fuzzy profile matching
  - Fuzzy role matching
  - Contact lookup by email
  - Account lookup by account number
  - Standard user provisioning into a community
  - Standard user login into a community
  - Default profile ID usage for portal Just-in-Time provisioning

## Configure SAML Settings for Single Sign-On

- Default portal role usage for portal Just-in-Time provisioning
- Username generation for portal Just-in-Time provisioning

For example, to support custom fields in the generated handler code, find the “Handle custom fields here” comment in the generated code. After that code comment, insert your custom field code. For more information and examples, see the [SamlJitHandler Interface documentation](#).

 **Note:** If your identity provider sends JIT attributes for the Contact or Account object with the User object in the same assertion, the generated handler might not be able to make updates. For a list of User fields that cannot be updated at the same time as the Contact or Account fields, see [sObjects That Cannot Be Used Together in DML Operations](#).

## Test the single sign-on connection

---

After you have configured and saved your SAML settings, test them by trying to access the identity provider's application. Your identity provider directs the user's browser to POST a form containing SAML assertions to the Salesforce login page. Each assertion is verified, and if successful, single sign-on is allowed.

If you have difficulty signing on using single sign-on after you have configured and saved your SAML settings, use the [SAML Assertion Validator](#). You might have to obtain a SAML assertion from your identity provider first.

If your users are having problems using SAML to log in, you can [review the SAML login history](#) to determine why they were not able to log in and share that information with your identity provider.

If you are using SAML version 2.0, after you've finished configuring SAML, the OAuth 2.0 Token Endpoint field is populated. Use the token with the web single sign-on authentication flow for OAuth 2.0.

## Working With Your Identity Provider


1. You must gather the following information from your identity provider before configuring Salesforce for SAML.


- The version of SAML the identity provider uses (1.1 or 2.0)
- The entity ID of the identity provider (also known as the issuer)
- An authentication certificate.

 **Tip:** Be sure to store the certificate where you can access it from your browser. This will be uploaded to Salesforce in a later step.


• The following SAML assertion parameters, as appropriate:

- The SAML user ID type
- The SAML user ID location
- Attribute Name
- Attribute URI
- Name ID format

 **Note:** Attribute Name, Attribute URI, and Name ID format are only necessary if the [SAML User ID Location](#) is in an Attribute element, and not the name identifier element of a Subject statement.

 **Tip:** To set up single sign-on quickly, you can import SAML 2.0 settings from an XML file (or a URL pointing to the file) on the Single Sign-On Settings page. Obtain the XML from your identity provider.

You may also want to share [more information](#) about these values with your identity provider.

 **Tip:** Enable Salesforce for SAML and take a screenshot of the page for your identity provider. From Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**, click **Edit**, then select **SAML Enabled**.

2. Work with your identity provider to setup the [start, login, and logout pages](#).
3. Share the [example SAML assertions](#) with your identity provider so they can determine the format Salesforce requires for successful single sign-on.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application

AND

Modify All Data

## Identity Provider Values

Before you can configure Salesforce for SAML, you must receive information from your identity provider. This information must be used on the [single sign-on page](#).

The following information might be useful for your identity provider.

Field	Description
SAML Version	The version of SAML your identity provider uses. Salesforce currently supports version 1.1 and 2.0. The SAML specifications for the various versions are linked below: <ul style="list-style-type: none"> <li>• <a href="#">SAML 1.1</a></li> <li>• <a href="#">SAML 2.0</a></li> </ul>
Issuer	The issuer in SAML requests generated by Salesforce, and is also the expected audience of any inbound SAML Responses. If you don't have <a href="#">domains</a> deployed, this value is always <code>https://saml.salesforce.com</code> . If you have domains deployed, Salesforce recommends that you use your custom domain name. You can find the value on the Single Sign-On Settings page. From Setup, enter <i>Single Sign-On Settings</i> in the Quick Find box, then select <b>Single Sign-On Settings</b> .
Entity ID	The Entity ID—a URL that uniquely identifies your SAML identity provider. SAML assertions sent to Salesforce must match this value exactly in the <code>&lt;saml:Issuer&gt;</code> attribute of SAML assertions.
Identity Provider Certificate	The authentication certificate issued by your identity provider.
Request Signing Certificate	The certificate (saved in the Certificate and Key Management page in Setup) used to generate the signature on a SAML request to the identity provider when Salesforce is the service provider for a service provider-initiated SAML login. If a certificate has not been saved in the Certificate and Key Management page in Setup, Salesforce uses the global proxy certificate by default. Using a saved signing certificate provides more control over events, such as certificate expiration, than using the global proxy certificate.
Request Signature Method	The hashing algorithm for encrypted requests, either <code>RSA-SHA1</code> or <code>RSA-SHA256</code> .
SAML Identity Type	The element in a SAML assertion that contains the string that identifies a Salesforce user. Values are: <p><b>Assertion contains User's Salesforce username</b></p> Use this option if your identity provider passes the Salesforce username in SAML assertions.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions


### USER PERMISSIONS


To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application AND Modify All Data

Field	Description
	<p><b>Assertion contains the Federation ID from the User object</b> Use this option if your identity provider passes an external user identifier, for example an employee ID, in the SAML assertion to identify the user.</p> <p><b>Assertion contains the User ID from the User object</b> Use this option if your identity provider passes an internal user identifier, for example a user ID from your Salesforce organization, in the SAML assertion to identify the user.</p>
SAML Identity Location	<p>The location in the assertion where a user should be identified. Values are:</p> <p><b>Identity is in the NameIdentifier element of the Subject statement</b> The Salesforce Username or FederationIdentifier is located in the &lt;Subject&gt; statement of the assertion.</p> <p><b>Identity is in an Attribute element</b> The Salesforce Username or FederationIdentifier is specified in an &lt;AttributeValue&gt;, located in the &lt;Attribute&gt; of the assertion.</p>
Attribute Name	If “Identity is in an Attribute element” is selected, this contains the value of the AttributeName that is specified in <Attribute> that contains the User ID.
Attribute URI	If SAML 1.1 is the specified SAML version and “Identity is in an Attribute element” is selected, this contains the value of the AttributeNamespace that is specified in <Attribute>.
Name ID Format	If SAML 2.0 is the specified SAML version and “Identity is in an Attribute element” is selected, this contains the value for the nameid-format. Possible values include unspecified, emailAddress or persistent. All legal values can be found in the “Name Identifier Format Identifiers” section of the <a href="#">Assertions and Protocols SAML 2.0 specification</a> .
Service Provider Initiated Request Binding	<p>If you’re using My Domain, chose the binding mechanism your identity provider requests for your SAML messages. Values are:</p> <p><b>HTTP POST</b> HTTP POST binding sends SAML messages using base64-encoded HTML forms.</p> <p><b>HTTP Redirect</b> HTTP Redirect binding sends base64-encoded and URL-encoded SAML messages within URL parameters.</p> <p>No matter what request binding is selected, the SAML Response will always use HTTP POST binding.</p>
Identity Provider Login URL	<p>For SAML 2.0 only: The URL where Salesforce sends a SAML request to start the login sequence.</p> <p>If you have <a href="#">domains</a> deployed and a value specified for this field, login requests are usually sent to the address specified by this field. However, if you need to bypass this value (for example, your identity provider is down) add the login parameter to the query string for the login page. For example: <code>http://mydomain.my.salesforce.com?login</code>.</p> <p> <b>Note:</b> This field appears in Developer Edition production and sandbox organizations by default and in production organizations only if My Domain is enabled. This field does not appear in trial organizations or sandboxes linked to trial organizations.</p>


Field	Description
Identity Provider Logout URL	For SAML 2.0 only: The URL to direct the user to when they click the <b>Logout</b> link in Salesforce. The default is <code>http://www.salesforce.com</code> .   <b>Note:</b> This field appears in Developer Edition production and sandbox organizations by default and in production organizations only if My Domain is enabled. This field does not appear in trial organizations or sandboxes linked to trial organizations.
Salesforce Login URL	The URL associated with logging in for the Web browser single sign-on flow.
OAuth 2.0 Token Endpoint	For SAML 2.0 only: The ACS URL used with the API when enabling Salesforce as an identity provider in the Web single sign-on OAuth assertion flow.
Custom Error URL	The URL of the page users should be directed to if there's an error during SAML login. It must be a publicly accessible page, such as a public site Visualforce page. The URL can be absolute or relative.

## Start, Login, and Logout URL Values

In addition to the information used during the single sign-on, your identity provider can also set the start, login, and logout pages. You can also specify these pages yourself when you [configure single sign-on](#).

The following information might be useful to your identity provider if they are setting these pages.

- The SAML specification supports an HTML form that is used to pass the SAML assertion via HTTPS POST.
- For SAML 1.1, the SAML identity provider can embed name-value pairs in the `TARGET` field to pass this additional information to Salesforce prepended with a specially formatted URL that contains URL-encoded parameters.
- The URL for SAML 1.1 to include in the `TARGET` field is as follows: `https://saml.salesforce.com/?`
- For SAML 2.0, instead of using the `TARGET` field, the identity providers uses the `<AttributeStatement>` in the SAML assertion to specify the additional information.
- Salesforce supports the following parameters:

-  **Note:** For SAML 1.1 these parameters must be URL-encoded. This allows the URLs, passed as values that include their own parameters, to be handled correctly. For SAML 2.0, these parameters are part of the `<AttributeStatement>`.
  - `ssoStartPage` is the page to which the user should be redirected when trying to log in with SAML. The user is directed to this page when requesting a protected resource in Salesforce, without an active session. The `ssoStartPage` should be the SAML identity provider's login page.
  - `startURL` is the URL where you want the user to be directed when sign-on completes successfully. This URL can be absolute, such as `https://yourInstance.salesforce.com/001/o` or it can be relative, such as `/001/o`. This parameter is only used in SAML 1.1. In SAML 2.0, the start URL is the page the user attempted to access before they were authenticated.
  - `logoutURL` is the URL where you want the user to be directed when they click the **Logout** link in Salesforce. The default is `http://www.salesforce.com`.

The following sample `TARGET` field is for SAML 1.1, and includes properly-encoded parameters. It passes a customized start page, as well as start and logout URLs embedded as parameter values in the query string.

```
https://saml.salesforce.com/?ssoStartPage=https%3A%2F%2Fwww.customer.org%2Flogin%2F&startURL=%2F001%2Fo&logoutURL=http%3A%2F%2Fwww.salesforce.com
```

The following is an example of an <AttributeStatement> for SAML 2.0 that contains both ssoStartPage and logoutURL:

```
<saml:AttributeStatement>
  <saml:Attribute Name="ssoStartPage"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:anyType">
    http://www.customer.org
  </saml:AttributeValue>
</saml:Attribute>

  <saml:Attribute Name="logoutURL"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
  <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
    https://www.salesforce.com
  </saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

## Customize SAML Start, Error, Login, and Logout Pages

You can customize the start, error, login, and logout pages for single sign-on users using SAML 1.1 or 2.0. As part of your configuration, decide the following:

- If your identity provider uses SAML 1.1, the URL to direct the user to when single sign-on successfully completes (known as the start page). This URL can be absolute, such as `https://yourInstance.salesforce.com/001/o` or it can be relative, such as `/001/o`. This URL must be an endpoint that accepts SAML authentication requests.  
In SAML 2.0, the start page is the page the user attempted to access before they were authenticated. The SAML 2.0 start page must support Sp-init single sign-on.  
If you are using SAML 2.0, you can also use the `RelayState` parameter to control where users get redirected after a successful login.
- The single sign-on start page where Salesforce sends a SAML request to start the login sequence.  
We recommend that if you specify a single sign-on start page that you also specify a logout page. When you specify a logout page, when a user clicks logout or if a user's session expires, the user is redirected to that page. If you don't specify a logout page, the user is redirected to the general Salesforce login page.
- The URL to direct the user to when they click the Logout link in Salesforce (known as the logout page). The default is `https://login.salesforce.com`, unless MyDomain is enabled. If My Domain is enabled, the default is `https://customdomain.my.salesforce.com`.

For SAML 2.0, these values can be set either during the single sign-on configuration, or by your identity provider in the login URL or SAML assertion. The order of precedence is:

1. Session cookie—if you've already logged in to Salesforce and a cookie still exists, the login and logout pages specified by the session cookie are used.
2. Values passed in from the identity provider.
3. Values from the single sign-on configuration page.

If you decide not to add these values to the single sign-on configuration, share them with your identity provider. The identity provider must [use these values](#) either in the login URL or the assertion.

You can also decide if you want users to be directed to a custom error page if there's an error during SAML login: It must be a publicly accessible page, such as a public site Visualforce page. The URL can be absolute or relative. Use this value when you [configure SAML](#).

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data



## Example SAML Assertions

Share the example SAML assertions with your identity provider so they can determine the format of the information Salesforce requires for successful single-sign on. The assertion must be signed according to the [XML Signature specification](#), using RSA and either SHA-1 or SHA-256.

In addition to the general single sign-on examples for both SAML 1.1 and SAML 2.0, use the following samples for the specific feature:

- [assertions for portals](#)
- [assertions for Sites](#)
- [SOAP message for delegated authentication](#)
- [assertion for just-in-time provisioning](#)

**SAML User ID type is the Salesforce username, and SAML User ID location is the <NameIdentifier> element in the <Subject> element**

SAML 1.1:

```
<Subject>
  <NameIdentifier>user101@salesforce.com</NameIdentifier>
</Subject>
```

SAML 2.0:

```
<saml:Subject>
  <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">user101@salesforce.com</saml:NameID>

  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2008-06-26T02:44:24.173Z"
Recipient="http://localhost:9000"/>
  </saml:SubjectConfirmation>
</saml:Subject>
```

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application

AND

Modify All Data

**SAML User ID type is the Salesforce username, and SAML User ID location is the <Attribute> element**

SAML 1.1:

```
<AttributeStatement>
  <Subject>
    <NameIdentifier>this value doesn't matter</NameIdentifier>
    <SubjectConfirmation>
      <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod>

    </SubjectConfirmation>
  </Subject>
  <Attribute AttributeName="MySfdcName" AttributeNamespace="MySfdcURI">
    <AttributeValue>user101@salesforce.com</AttributeValue>
  </Attribute>
</AttributeStatement>
```

SAML 2.0:

```
<saml:AttributeStatement>
  <saml:Attribute FriendlyName="fooAttrib" Name="SFDC_USERNAME"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
    user101@salesforce.com
  </saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```


**SAML User ID type is the Salesforce User object's FederationIdentifier field, and SAML User ID location is the <NameIdentifier> element in the <Subject> element**

SAML 1.1:

```
<AttributeStatement>
  <saml:Subject>
    <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.0:assertion"
NameQualifier="www.saml_assertions.com">
      MyName
    </saml:NameIdentifier>
  </saml:Subject>
</AttributeStatement>
```

SAML 2.0:

```
<saml:Subject>
  <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">MyName</saml:NameID>
  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2008-06-26T02:48:25.730Z"
Recipient="http://localhost:9000/">
  </saml:SubjectConfirmation>
</saml:Subject>
```

 **Note:** The name identifier can be any arbitrary string, including email addresses or numeric ID strings.

**SAML User ID type is the Salesforce User object's `FederationIdentifier` field, and SAML User ID location is the `<Attribute>` element**

SAML 1.1:

```
<AttributeStatement>
  <Subject>
    <NameIdentifier>who cares</NameIdentifier>
    <SubjectConfirmation>
      <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod>

    </SubjectConfirmation>
  </Subject>
  <Attribute AttributeName="MyName" AttributeNamespace="MyURI">
    <AttributeValue>user101</AttributeValue>
  </Attribute>
</AttributeStatement>
```

SAML 2.0:

```
<saml:AttributeStatement>
  <saml:Attribute FriendlyName="fooAttrib" Name="SFDC_ATTR"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
    user101
  </saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

**SAML User ID type is the Salesforce username, and SAML User ID location is the `<NameIdentifier>` element in the `<Subject>` element**

The following is a complete SAML response for SAML 2.0:

```
<samlp:Response ID="_257f9d9e9fa14962c0803903a6ccad931245264310738"
IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
    https://www.salesforce.com
  </saml:Issuer>

  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>

  <saml:Assertion ID="_3c39bc0fe7b13769cab2f6f45eba801b1245264310738"
IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">
    <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
      https://www.salesforce.com
    </saml:Issuer>

    <saml:Signature>
      <saml:SignedInfo>
        <saml:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <saml:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
```

```

    <saml:Reference URI="#_3c39bc0fe7b13769cab2f6f45eba801b1245264310738">
      <saml:Transforms>
        <saml:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        <saml:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <ec:InclusiveNamespaces PrefixList="ds saml xs"/>
        </saml:Transform>
      </saml:Transforms>
      <saml:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <saml:DigestValue>vzR9Hfp8d16576tEDeq/zhpmLoo=
      </saml:DigestValue>
    </saml:Reference>
  </saml:SignedInfo>
  <saml:SignatureValue>
    AzID5hhJeJlG21lUDvZswNurlrPtR7S37QYH2W+Unln8c6kTC
    Xr/lihEKpCA2PZt86eBntFBVDWTRlh/W3yUgGOqQBjMFOVbhK
    M/CbLHbBUVT5TcxIqvsNvIFdjIGNkf1W0SBqRKZQJ6tzxCcLo
    9dXqAyAUkqDpX5+AyIwtwrdCPNmncUM4dtRPjI05CL1rRaGeyX
    3kkqOL8p0vjm0fazU5tCAJLbYuYgU1LivPSahWNcprRS1CI4e
    Pn2oiVDyrcc4et12inPMTc2lGIWWWWJyHOPSiXRSkEAIwQVjf
    Qm5cpli44Pv8FCrdGWpEE0yXsPBvDkM9jIzwCYGG2fKaLBag==
  </saml:SignatureValue>
  <saml:KeyInfo>
    <saml:X509Data>
      <saml:X509Certificate>
        MIIEATCCAumgAwIBAgIBBTANBgkqhkiG9w0BAQ0FADCBgzELM
        [Certificate truncated for readability...]
      </saml:X509Certificate>
    </saml:X509Data>
  </saml:KeyInfo>
</saml:Signature>

<saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
    saml01@salesforce.com
  </saml:NameID>

  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2009-06-17T18:50:10.738Z"
      Recipient="https://login.salesforce.com"/>
    </saml:SubjectConfirmation>
  </saml:Subject>

  <saml:Conditions NotBefore="2009-06-17T18:45:10.738Z"
    NotOnOrAfter="2009-06-17T18:50:10.738Z">

    <saml:AudienceRestriction>
      <saml:Audience>https://saml.salesforce.com</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>

  <saml:AuthnStatement AuthnInstant="2009-06-17T18:45:10.738Z">
    <saml:AuthnContext>
      <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified

```

```

        </saml:AuthnContextClassRef>
    </saml:AuthnContext>
</saml:AuthnStatement>

<saml:AttributeStatement>

    <saml:Attribute Name="portal_id">
        <saml:AttributeValue xsi:type="xs:anyType">060D00000000SHZ
        </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="organization_id">
        <saml:AttributeValue xsi:type="xs:anyType">00DD0000000F7L5
        </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="ssostartpage"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

        <saml:AttributeValue xsi:type="xs:anyType">
            http://www.salesforce.com/security/saml/saml20-gen.jsp
        </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="logouturl"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

        <saml:AttributeValue xsi:type="xs:string">
            http://www.salesforce.com/security/del_auth/SsoLogoutPage.html
        </saml:AttributeValue>
    </saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>

```

## Sample SAML Assertions for Portals

The following shows the `portal_id` and `organization_id` attributes in a SAML assertion statement:

```

<saml:AttributeStatement>
    <saml:Attribute Name="portal_id">
        <saml:AttributeValue xsi:type="xs:anyType">060D00000000SHZ</saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="organization_id">
        <saml:AttributeValue xsi:type="xs:anyType">00DD0000000F7P5</saml:AttributeValue>

    </saml:Attribute>
</saml:AttributeStatement>

```

The following is a complete SAML assertion statement that can be used for single sign-on for portals. The organization is using federated sign-on, which is included in an attribute (see the `<saml:AttributeStatement>` in bold text in the assertion), not in the subject.

```
<samlp:Response ID="_f97faa927f54ab2c1fef230eee27cba21245264205456"
  IssueInstant="2009-06-17T18:43:25.456Z" Version="2.0">
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
    https://www.salesforce.com</saml:Issuer>

  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>

  <saml:Assertion ID="_f690da2480a8df7fcc1cbee5dc67dbbb1245264205456"
    IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">
    <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
      https://www.salesforce.com
    </saml:Issuer>

    <saml:Signature>
      <saml:SignedInfo>
        <saml:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <saml:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />

        <saml:Reference URI="#_f690da2480a8df7fcc1cbee5dc67dbbb1245264205456">
          <saml:Transforms>
            <saml:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <saml:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <ec:InclusiveNamespaces PrefixList="ds saml xs" />
            </saml:Transform>
          </saml:Transforms>
          <saml:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <saml:DigestValue>vzR9Hfp8d16576tEDeq/zhpmLoo=
          </saml:DigestValue>
        </saml:Reference>
      </saml:SignedInfo>
      <saml:SignatureValue>
        AzID5hhJeJlG2llUDvZswNUrlrPtR7S37QYH2W+Un1n8c6kTC
        Xr/lihEKPCa2PZt86eBntFBVDWTRlh/W3yUgGOqQBjMFOVbhK
        M/CbLHbBUVT5TcxIqvsNvIFdjIGNkflW0SBqRKZOJ6tzxCcLo
        9dXqAyAUkqDpX5+AyItwrdCPNmncUM4dtRPjI05CLlrRaGeyX
        3kkqOL8p0vjm0fazU5tCAJLbYuYgU1LivPSahWNcpvRS1CI4e
        Pn2oiVDyrcc4et12inPMTc2lGIWWWWJyHOPSiXRSkEAIwQVjf
        Qm5cpli44Pv8FCrdGWpEE0yXsPBvDkM9jIzwCYGG2fKaLBag==
      </saml:SignatureValue>
      <saml:KeyInfo>
        <saml:X509Data>
          <saml:X509Certificate>
            MIIIEATCCAumgAwIBAgIBBTANBgkqhkiG9w0BAQ0FADCBgzELM
            Certificate truncated for readability...
          </saml:X509Certificate>
        </saml:X509Data>
      </saml:KeyInfo>
    </saml:Signature>
  </saml:Assertion>
</samlp:Response>
```

```

<saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">null

  </saml:NameID>

  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2009-06-17T18:48:25.456Z"
      Recipient="https://login.salesforce.com/?saml=02HKiPoin4f49GRMsOdFmhTgi
        _0nR7BBAflopdnD3gtixujECWpxr9klAw"/>
    </saml:SubjectConfirmation>
  </saml:Subject>

  <saml:Conditions NotBefore="2009-06-17T18:43:25.456Z"
    NotOnOrAfter="2009-06-17T18:48:25.456Z">

    <saml:AudienceRestriction>
      <saml:Audience>https://saml.salesforce.com</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>

  <saml:AuthnStatement AuthnInstant="2009-06-17T18:43:25.456Z">

    <saml:AuthnContext>
      <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified

      </saml:AuthnContextClassRef>
    </saml:AuthnContext>
  </saml:AuthnStatement>

  <saml:AttributeStatement>

    <saml:Attribute FriendlyName="Friendly Name" Name="federationId"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml:AttributeValue xsi:type="xs:string">saml_portal_user_federation_id
      </saml:AttributeValue>
      <saml:AttributeValue xsi:type="xs:string">SomeOtherValue
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="portal_id">
      <saml:AttributeValue xsi:type="xs:anyType">060D00000000SHZ
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="organization_id">
      <saml:AttributeValue xsi:type="xs:anyType">00DD00000000F7Z5
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="ssostartpage"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">

      <saml:AttributeValue xsi:type="xs:anyType">

```

```

        http://www.salesforce.com/qa/security/saml/saml20-gen.jsp
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="logouturl"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

    <saml:AttributeValue xsi:type="xs:string">
        http://www.salesforce.com/qa/security/del_auth/SsoLogoutPage.html
    </saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>

```

## Sample SAML Assertion for Sites

The following shows the `portal_id`, `organization_id`, and `siteurl` attributes in a SAML assertion statement:

```

<saml:AttributeStatement>
  <saml:Attribute Name="portal_id">
    <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:anyType">060900000004cDk
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute Name="organization_id">
    <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:anyType">00D900000008bX0
    </saml:AttributeValue></saml:Attribute>
  <saml:Attribute Name="siteurl">
    <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:anyType">https://apl.force.com/mySuffix</saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>

```

## Sample SOAP Message for Delegated Authentication

As part of the delegated authentication single sign-on process, a Salesforce server makes a SOAP 1.1 request to authenticate the user who is passing in the credentials. Here is an example of this type of request. Your single sign-on Web service needs to accept this request, process it, and return a true or false response.

### Sample Request

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <Authenticate xmlns="urn:authentication.soap.sforce.com">
      <username>sampleuser@sample.org</username>
      <password>myPassword99</password>
    </Authenticate>
  </soapenv:Body>
</soapenv:Envelope>

```



```

    <sourceIp>1.2.3.4</sourceIp>
  </Authenticate>
</soapenv:Body>
</soapenv:Envelope>

```

### Sample Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <AuthenticateResult xmlns="urn:authentication.soap.sforce.com">
      <Authenticated>false</Authenticated>
    </AuthenticateResult>
  </soapenv:Body>
</soapenv:Envelope>

```

## Sample SAML Assertion for Just-In-Time Provisioning

The following is a sample SAML assertion for just in time provisioning.

```

<saml:AttributeStatement>

  <saml:Attribute Name="User.Username"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="User.Phone"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">415-123-1234
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="User.FirstName"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">Testuser
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="User.LanguageLocaleKey"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">en_US
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="User.CompanyName"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">Salesforce.com
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="User.Alias"

```

```
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">tlee2
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.CommunityNickname"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">tlee2
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.UserRoleId"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">0000000000000000
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.Title"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">Mr.
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.LocaleSidKey"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">en_CA
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.Email"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">testuser@salesforce.com
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name=" User.FederationIdentifier"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">tlee2
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.TimeZoneSidKey"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">America/Los_Angeles
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.LastName"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">Lee
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.ProfileId"
```

```
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">00ex0000001pBNL
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.IsActive"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">1
    </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.EmailEncodingKey"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">UTF-8
    </saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

# VIEW AND EDIT SINGLE SIGN-ON SETTINGS

After you've configured your Salesforce org to use SAML, you can manage the SAML configuration from the Single Sign-On Settings page.

From Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**.

After the SAML configuration completes, the Single Sign-On Settings page displays the generated URLs and OAuth 2.0 token endpoint.

Field	Description
Salesforce Login URL	For SAML 2.0. The URL associated with the login for the Web SSO OAuth assertion flow. This URL appears if you configured SAML with "Assertion contains the User's Salesforce username" for SAML Identity Type and "Identity is in the NameIdentifier element of the Subject statement" for SAML Identity Location.
Salesforce Logout URL	For SAML 2.0. The Salesforce logout URL that users are directed to after they log off. This URL appears if you didn't specify a value for Identity Provider Logout URL.
OAuth 2.0 Token Endpoint	For SAML 2.0. The ACS URL used when enabling Salesforce as an identity provider in the Web SSO OAuth assertion flow.

From this page you can do any of the following:

- Click **Edit** to change the existing SAML configuration.
- Click **SAML Assertion Validator** to validate the SAML settings for your org using a SAML assertion provided by your identity provider.
- Click **Download Metadata** to download an XML file of your SAML configuration settings to send to your identity provider. The identity provider can then upload these configuration settings to connect to your Salesforce orgcommunity. Enabled only if your identity provider supports metadata and if you are using SAML 2.0.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application
- AND
- Modify All Data

# VALIDATING SAML SETTINGS FOR SINGLE SIGN-ON

If your users have difficulty logging into Salesforce after you [configure Salesforce for single sign-on](#), use the SAML Assertion Validator and the [login history](#) to validate the SAML assertions sent by your identity provider.

1. Obtain a SAML assertion from your identity provider. The assertion can be either in plain XML format or base64 encoded.

If a user tries to log in to Salesforce and fails, the invalid SAML assertion is used to automatically populate the SAML Assertion Validator if possible.

2. From Setup, enter *Single Sign-On Settings* in the **Quick Find** box, then select **Single Sign-On Settings**, then click **SAML Assertion Validator**.
3. Enter the SAML assertion into the text box, and click **Validate**.
4. Share the results of the [validation errors](#) with your identity provider.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data

## SAML Assertion Validation Errors

Salesforce imposes the following validity requirements on assertions:

### Authentication Statement

The identity provider must include an `<AuthenticationStatement>` in the assertion.

### Conditions Statement

If the assertion contains a `<Conditions>` statement, it must contain a valid timestamp.

### Timestamps

The validity period specified in an assertion is honored. In addition, an assertion's timestamp must be less than five minutes old, plus or minus three minutes, regardless of the assertion's validity period setting. This allows for differences between machines. The `NotBefore` and `NotOnOrAfter` constraints must also be defined and valid.

### Attribute

If your Salesforce configuration is set to `Identity is in an Attribute` element, the assertion from the identity provider must contain an `<AttributeStatement>`.

If you are using SAML 1.1, both `<AttributeName>` and `<AttributeNamespace>` are required as part of the `<AttributeStatement>`.

If you are using SAML 2.0, only `<AttributeName>` is required.

### Format

The `Format` attribute of an `<Issuer>` statement must be set to

`"urn:oasis:names:tc:SAML:2.0:nameid-format:entity"` or not set at all.

For example:

```
<saml:Issuer
  Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">https://www.salesforce.com</saml:Issuer>
```

The following example is also valid:

```
<saml:Issuer >https://www.salesforce.com</saml:Issuer>
```

### Issuer

The issuer specified in an assertion must match the issuer specified in Salesforce.

### Subject

The subject of the assertion must be resolved to be either the Salesforce username or the Federation ID of the user.

### Audience

The `<Audience>` value is required and must match the `Entity ID` from the single sign-on configuration. The default value is `https://saml.salesforce.com`.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application

AND

Modify All Data

**Recipient**

The recipient specified in an assertion must match either the Salesforce login URL specified in the Salesforce configuration or the OAuth 2.0 token endpoint. This is a required portion of the assertion and is always verified.

**Signature**

A valid signature must be included in the assertion. The signature must be created using the private key associated with the certificate that was provided in the SAML configuration.

**Recipient**

Verifies that the recipient and organization ID received in the assertion matches the expected recipient and organization ID, as specified in the single sign-on configuration. This is an optional portion of the assertion and is only verified if it's present. For example:

```
Recipient that we found in the assertion: http://aalbert-salesforce.com:8081/  
?saml=02HKiPoin4zeKLPYxfj3twkPsNSJF3fxsh0Jnq4vVeQr3xNkIWmZC_IVk3  
Recipient that we expected based on the Single Sign-On Settings page:  
http://asmith.salesforce.com:8081/  
?saml=EK03Almz90Cik_ig0L97.0BRme6mT4o6nzi0t_JROL6HLbdR1WVP5aQO5w  
Organization Id that we expected: 00Dx0000000BQ1I  
Organization Id that we found based on your assertion: 00D000000000062
```

**Site URL Attribute**

Verifies if a valid Sites URL is provided. Values are:

- Not Provided
- Checked
- Site URL is invalid
- HTTPS is required for Site URL
- The specified Site is inactive or has exceeded its page limit

# REVIEWING THE SAML LOGIN HISTORY

When a user logs in to Salesforce from another application using single sign-on, SAML assertions are sent to the Salesforce login page. The assertions are checked against assertions in the authentication certificate that are specified on the Single Sign-On Settings page in Setup. If a user fails to log in, a message is written to the login history log that indicates why the login failed. In addition, the [SAML Assertion Validator](#) may be automatically populated with the invalid assertion.

To view the login history, from Setup, enter *Login History* in the *Quick Find* box, then select **Login History**. After viewing the login history, you may want to share the information with your identity provider.

The following are the possible failures:

## Assertion Expired

An assertion's [timestamp](#) is more than five minutes old.



**Note:** Salesforce does make an allowance of three minutes for clock skew. This means, in practice, that an assertion can be as much as eight minutes after the timestamp time, or three minutes before it. This amount of time may be less if the assertion's validity period is less than five minutes.

## Assertion Invalid

An assertion is not valid. For example, the `<Subject>` element of an assertion might be missing.

## Audience Invalid

The value specified in `<Audience>` must be `https://saml.salesforce.com`.

## Configuration Error/Perm Disabled

Something is wrong with the SAML configuration in Salesforce. For example, the uploaded certificate might be corrupted, or the organization preference might have been turned off. To check your configuration, from Setup, enter *Single Sign-On Settings* in the *Quick Find* box, then select **Single Sign-On Settings**. Next, get a sample SAML assertion from your identity provider, and then click [SAML Assertion Validator](#).

## Issuer Mismatched

The issuer or entity ID specified in an assertion does not match the issuer specified in your Salesforce configuration.

## Recipient Mismatched

The recipient specified in an assertion does not match the recipient specified in your Salesforce configuration.

## Replay Detected

The same assertion ID was used more than once. [Assertion IDs](#) must be unique within an organization.

## Signature Invalid

The signature in an assertion cannot be validated by the certificate in your Salesforce configuration.

## Subject Confirmation Error

The `<Subject>` specified in the assertion does not match the SAML configuration in Salesforce.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application AND Modify All Data



# ABOUT JUST-IN-TIME PROVISIONING FOR SAML

With Just-in-Time provisioning, you can use a SAML assertion to create regular and portal users on the fly the first time they try to log in. This eliminates the need to create user accounts in advance. For example, if you recently added an employee to your organization, you don't need to manually create the user in Salesforce. When they log in with single sign-on, their account is automatically created for them, eliminating the time and effort with on-boarding the account. Just-in-Time provisioning works with your SAML identity provider to pass the correct user information to Salesforce in a SAML 2.0 assertion. You can both create and modify accounts this way. Because Just-in-Time provisioning uses SAML to communicate, your organization must have SAML-based single sign-on enabled.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All Editions**

## Benefits of Just-in-Time Provisioning

Implementing Just-in-Time provisioning can offer the following advantages to your organization.

- **Reduced Administrative Costs:** Provisioning over SAML allows customers to create accounts on-demand, as part of the single sign-on process. This greatly simplifies the integration work required in scenarios where users need to be dynamically provisioned, by combining the provisioning and single sign-on processes into a single message.
- **Increased User Adoption:** Users only need to memorize a single password to access both their main site and Salesforce. Users are more likely to use your Salesforce application on a regular basis.
- **Increased Security:** Any password policies that you have established for your corporate network are also in effect for Salesforce. In addition, sending an authentication credential that is only valid for a single use can increase security for users who have access to sensitive data.

## Just-in-Time Provisioning Requirements and SAML Assertion Fields

Just-in-Time provisioning requires the creation of a SAML assertion. Consider the following when creating your SAML assertion.

- `ProvisionVersion` is supported as an optional attribute. If it isn't specified, the default is 1.0. For example:

```
<saml:Attribute Name="ProvisionVersion" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">1.0</saml:AttributeValue>
</saml:Attribute>
```

- ProfileIDs change per organization, even for standard profiles. To make it easier to find the profile name, Salesforce allows you to do a profile name lookup by passing the `ProfileName` into the `ProfileId` field.

## Field Requirements for the SAML Assertion

To correctly identify which object to create in Salesforce, you must use the `User.` prefix for all fields passed in the SAML assertion. In this example, the `User.` prefix has been added to the `Username` field name.

```
<saml:Attribute
  Name="User.Username"
```

```
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org</saml:AttributeValue>
</saml:Attribute>
```

The following User fields are required:

**Email****LastName****ProfileId****Username (insert only)**

The following User fields are supported.

**FirstName****CommunityNickname****FederationIdentifier****TimeZoneSidKey**

This value is the Time Zone field on the User object.

**LanguageLocaleKey**

This value is the Language field on the User object.

**LocaleSidKey**

This value is the Locale field on the User object.

**EmailEncodingKey**

This value is the Email Encoding field on the User object.

**DefaultCurrencyIsoCode****Role****Alias****Title****Phone****CompanyName**

This value is the Company field on the User object.

**Active**

This value is the Active field on the User object and User.isActive in the API.

**AboutMe****Street**

This value is part of the Address compound field on the User object and User.Street in the API.

**State**

This value is part of the Address compound field on the User object and User.State in the API.

**City**

This value is part of the Address compound field on the User object and User.City in the API.

**Zip**

This value is part of the Address compound field on the User object and User.PostalCode in the API.

**Country**

This value is part of the Address compound field on the User object and User.Country in the API.

**ReceivesAdminInfoEmails****ForecastEnabled**

This value is the Allow Forecasting checkbox on the User object.

**CallCenter**

This value is the User.CallCenterId in the API.

**Manager****MobilePhone****DelegatedApproverId****Department****Division****EmployeeNumber****Extension****Fax****ReceivesInfoEmails**


Other field requirements:

- Only text type custom fields are supported.
- Only the `insert` and `update` functions are supported for custom fields.
- When using the API for user creation, you can pass the new username into the `User.Username` field. You can also specify the `User.FederationIdentifier` if it is present. However, the `Username` and `FederationIdentifier` fields can't be updated with API.

## Just-in-Time Provisioning and SAML Assertion Fields for Portals

---

With Just-in-Time (JIT) provisioning for portals, you can use a SAML assertion to create customer and partner portal users on the fly the first time they try to log in. This eliminates the need to create user accounts in advance. Because JIT uses SAML to communicate, your organization must have SAML-based single sign-on enabled.

 **Note:** Starting with Summer '13, Customer Portals and partner portals are no longer available for new organizations. Existing organizations continue to have access to these portals. If you don't have a portal, but want to easily share information with your customers or partners, try Communities.


Existing organizations using Customer Portals and partner portals may continue to use their portals or transition to Communities. Contact your Salesforce Account Executive for more information.

## Creating Portal Users

The `Portal ID` and `Organization ID` must be specified as part of the SAML assertion. You can find both of these on the company information page for the organization or portal. Because you can also provision regular users, the `Portal ID` is used to distinguish between a regular and portal JIT provisioning request. If no `Portal ID` is specified, then the request is treated as a JIT request for regular platform user. Here are the requirements for a creating a portal user.

- You must specify a `Federation ID`. If the ID belongs to an existing user account, the user account is updated. In case of an inactive user account, the user account is updated, but left inactive unless `User.IsActive` in the JIT assertion is set to true. If there is no user account with that `Federation ID`, the system creates a new user.

- If the portal isn't self-registration enabled and a default new user profile and role aren't specified, the `User.ProfileId` field must contain a valid profile name or ID associated with the portal. In addition, the `User.PortalRole` field must contain a valid portal role name or ID. Use `Worker` for all portal users.

 **Note:** The `User.Role` must be null.

## Creating and Modifying Accounts

Create or modify an account by specifying a valid `Account.Id` or both the `Account.AccountNumber` and `Account.Name`.

- Matching is based on `Account.AccountNumber`. If multiple accounts are found, an error is displayed. Otherwise, the account is updated.
- If no matching account is found, one is created.
- You must specify the `Account.Owner` in the SAML assertion and ensure that the field level security for the `Account.AccountNumber` field is set to visible for this owner's profile.

## Creating and Modifying Contacts

Create or modify a contact by specifying the a valid Contact ID in `User.Contact` or both the `Contact.Email` and `Contact.LastName`.

- Matching is based on `Contact.Email`. If multiple contacts are found, an error is displayed. Otherwise, the contact is updated.
- If no matching contact is found, one is created.

## Supported Fields for the Portal SAML Assertion

To correctly identify which object to create in Salesforce, you must use a prefix. In the SAML assertion, use the `Account` prefix for all fields in the Account schema (for example `Account.AccountId`) and `Contact` prefix for all fields in the Contact schema. In this example, the `Contact` prefix has been added to the `Email` field name.

```
<saml:Attribute
  Name="Contact.Email"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org</saml:AttributeValue>
</saml:Attribute>
```

In addition to the standard User attributes supported for regular SAML JIT users, these Account attributes are also supported. For example, specifying an `Account.Phone` attribute in the assertion will update the account's Phone field on the corresponding Account object.

**Name**

**AccountNumber**

**BillingCity**

**BillingCountry**

**BillingPostalCode**

**BillingState**

**BillingStreet**

**Owner**

The Owner field on the Account object is `Account.OwnerId` in the API.

**AnnualRevenue**

**Description**

**NumberOfEmployees**

**Fax**

**Industry**

**Ownership**

**Phone**

**Rating**

**ShippingAddress**

The Shipping Address field is a compound field.

**ShippingCity**

**ShippingCountry**

**ShippingPostalCode**

**ShippingState**

**ShippingStreet**

**Sic**

**TickerSymbol**

**Website**

These Contact attributes are supported.

**Account**

This value is the Account Name field on the Contact object and Account.Name in the API.

**Email**

**FirstName**

**LastName**

**Phone**

**CanAllowPortalSelfReg**

**AssistantName**

**AssistantPhone**

**Birthdate**

**Owner**

This value is the Contact Owner field on the Contact object and Contact.OwnerId in the API.

**Department**

**Description**

**DoNotCall**

**HasOptedOutOfEmail**

**Fax**

**HasOptedOutOfFax**

**HomePhone**

**LastCUUpdateDate**

This value is the Last Modified By field on the Contact object and Contact.LastModifiedDate in the API.

**LeadSource**

**MailingAddress**

The Mailing Address field is a compound field.

**MailingCity**

**MailingCountry**

**MailingPostalCode**

**MailingState**

**MailingStreet**

**MobilePhone**

**Salutation**

**OtherAddress**

The Other Address field is a compound field.

**OtherCity**

**OtherCountry**

**OtherPostalCode**

**OtherState**

**OtherStreet**

**OtherPhone**

**Title**

These additional User attributes are supported for portal users.

**AccountId**

**ContactId**

**PortalRole**

Use `worker` for all portal users.

## Just-in-Time Provisioning for Communities

---

With Just-in-Time (JIT) provisioning for Communities, you can use a SAML assertion to create customer and partner community users on the fly the first time they try to log in from an identity provider. This eliminates the need to create user accounts in advance. Because

JIT uses SAML to communicate, your organization must have SAML-based single sign-on enabled. Then, you can work with the identity provider to generate the necessary SAML assertions for JIT.

## SAML Single Sign-on Settings

Follow the instructions for [Configure SAML Settings for Single Sign-On](#) with `SAML Enabled`. Set the values for your configuration, as needed, and also include the following values specific to your community for JIT provisioning.

### 1. Check `User Provisioning Enabled`.



#### Note:

- Just-in-time provisioning requires a Federation ID in the user type. In `SAML User ID Type`, select `Assertion contains the Federation ID from the User object`.
- If your identity provider previously used the Salesforce username, communicate to them that they must use the Federation ID.

### 2. The **Entity ID** should be unique across your organization and begin with `https`. You can't have two SAML configurations with the same **Entity ID** in one organization. Specify whether you want to use the base domain (`https://saml.salesforce.com`) or the community URL (such as `https://acme.force.com/customers`) for the **Entity ID**. You must share this information with your identity provider.



**Tip:** Generally, use the community URL as the entity ID. If you are providing Salesforce to Salesforce services, you must specify the community URL.

### 3. In `SAML User ID Type`, select `Assertion contains the Federation ID from the User object`. If your identity provider previously used the Salesforce username, communicate to them that they must use the Federation ID.

## Creating and Modifying Community Users

The SAML assertion needs the following.

- A `Recipient URL`. This is the Community Login URL from the SAML Single Sign-On Settings detail page in your organization. The URL is in the following form.

```
https://<community_URL>/login?so=<orgID>
```

For example, `Recipient="https://acme.force.com/customers/login?so=00DD000000JsCM"` where `acme.force.com/customers` is the community home page and `00DD000000JsCM` is the `Organization ID`.

If an Assertion Decryption Certificate has been uploaded to the organization's SAML Single Sign-On Settings, include the certificate ID in the URL using the `sc` parameter, such as


```
Recipient="https://acme.force.com/customers/login?so=00DD000000JsCM&sc=0LE000000Dp"
```

where `0LE000000Dp` is the certificate ID.

- Salesforce attempts to match the `Federation ID` in the subject of the SAML assertion (or in an attribute element, depending upon how the SAML Identity Location is defined in the SAML Single Sign-On Settings) to the `User.FederationIdentifier` field of an existing user record.
  - If a matching user record is found, Salesforce uses the attributes in the SAML assertion to update the specified fields.
  - If a user with a matching user record isn't found, then Salesforce searches the contacts for a match based on the `Contact ID (User.Contact)` or email (`Contact.Email`). `Contact.Email` and `Contact.LastName` are both required properties when `User.Contact` is not specified, but matching is only based on `Contact.Email` when both properties exist.

- i. If a matching contact record is found, Salesforce uses the attributes in the SAML assertion to update the specified contact fields, and then inserts a new user record.
- ii. If a matching contact record isn't found, then Salesforce searches the accounts for a match based on the `Contact.Account` or `Contact.AccountNumber` specified in the SAML assertion. `Contact.AccountNumber` and `Contact.Name` are both required properties when `Contact.Account` is not specified, but matching is only based on `Contact.AccountNumber` when both properties exist.
  - i. If a matching account record is found, Salesforce inserts a new user record and updates the account records based the attributes provided in the SAML assertion.
  - ii. If a matching account record isn't found, Salesforce inserts new account, contact, and user records based on the attributes provided in the SAML assertion.

In the case of an inactive user account, the user account is updated, but left inactive unless `User.IsActive` in the JIT assertion is set to true. If there is no user account with that `FederationID`, the system creates a new user.

- If the community doesn't have self-registration enabled, and a default new user profile and role aren't specified, the `User.ProfileId` field must contain a valid profile name or ID associated with the community.
-  **Note:** Salesforce also supports custom fields on the User object in the SAML assertion. Any attribute in the assertion that starts with `User` is parsed as a custom field. For example, the attribute `User.NumberOfProductsBought__c` in the assertion is placed into the field `NumberOfProductsBought` for the provisioned user. Custom fields are not supported for Accounts or Contacts.

## Supported Fields for the Community SAML Assertion

To correctly identify which object to create in Salesforce, you must use a prefix. In the SAML assertion, use the `Account` prefix for all fields in the Account schema (for example `Account.AccountId`) and `Contact` prefix for all fields in the Contact schema. In this example, the `Contact` prefix has been added to the `Email` field name.

```
<saml:Attribute
  Name="Contact.Email"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org</saml:AttributeValue>
</saml:Attribute>
```

In addition to the standard User attributes supported for regular SAML JIT users, these Account attributes are also supported. For example, specifying an `Account.Phone` attribute in the assertion will update the account's Phone field on the corresponding Account object.

**Name**

**AccountNumber**

**BillingCity**

**BillingCountry**

**BillingPostalCode**

**BillingState**

**BillingStreet**

**Owner**

The Owner field on the Account object is `Account.OwnerId` in the API.



**AnnualRevenue**

**Description**

**NumberOfEmployees**

**Fax**

**Industry**

**Ownership**

**Phone**

**Rating**

**ShippingAddress**

The Shipping Address field is a compound field.

**ShippingCity**

**ShippingCountry**

**ShippingPostalCode**

**ShippingState**

**ShippingStreet**

**Sic**

**TickerSymbol**

**Website**

These Contact attributes are supported.

**Account**

This value is the Account Name field on the Contact object and Account.Name in the API.

**Email**

**FirstName**

**LastName**

**Phone**

**CanAllowPortalSelfReg**

**AssistantName**

**AssistantPhone**

**Birthdate**

**Owner**

This value is the Contact Owner field on the Contact object and Contact.OwnerId in the API.

**Department****Description****DoNotCall****HasOptedOutOfEmail****Fax****HasOptedOutOfFax****HomePhone****LastCUUpdatedDate**

This value is the Last Modified By field on the Contact object and Contact.LastModifiedDate in the API.

**LeadSource****MailingAddress**

The Mailing Address field is a compound field.

**MailingCity****MailingCountry****MailingPostalCode****MailingState****MailingStreet****MobilePhone****Salutation****OtherAddress**

The Other Address field is a compound field.

**OtherCity****OtherCountry****OtherPostalCode****OtherState****OtherStreet****OtherPhone****Title**


## Just-in-Time Provisioning Errors

---

Following are the error codes and descriptions for Just-in-Time provisioning for SAML.

SAML errors are returned in the URL parameter, for example:

```
http://login.salesforce.com/identity/jit/saml-error.jsp?
ErrorCode=5&ErrorDescription=Unable+to+create+user&ErrorDetails=
INVALID_OR_NULL_FOR_RESTRICTED_PICKLIST+TimeZoneSidKey
```

 **Note:** Salesforce redirects the user to a custom error URL if one is specified in your SAML configuration.

## Error Messages

Code	Description	Error Details
1	Missing Federation Identifier	MISSING_FEDERATION_ID
2	Mis-matched Federation Identifier	MISMATCH_FEDERATION_ID
3	Invalid organization ID	INVALID_ORG_ID
4	Unable to acquire lock	USER_CREATION_FAILED_ON_UROG
5	Unable to create user	USER_CREATION_API_ERROR
6	Unable to establish admin context	ADMIN_CONTEXT_NOT_ESTABLISHED
8	Unrecognized custom field	UNRECOGNIZED_CUSTOM_FIELD
9	Unrecognized standard field	UNRECOGNIZED_STANDARD_FIELD
11	License limit exceeded	LICENSE_LIMIT_EXCEEDED
12	Federation ID and username do not match	MISMATCH_FEDERATION_ID_AND_USERNAME_ATTRS
13	Unsupported provision API version	UNSUPPORTED_VERSION
14	Username change isn't allowed	USER_NAME_CHANGE_NOT_ALLOWED
15	Custom field type isn't supported	UNSUPPORTED_CUSTOM_FIELD_TYPE
16	Unable to map a unique profile ID for the given profile name	PROFILE_NAME_LOOKUP_ERROR
17	Unable to map a unique role ID for the given role name	ROLE_NAME_LOOKUP_ERROR
18	Invalid account	INVALID_ACCOUNT_ID
19	Missing account name	MISSING_ACCOUNT_NAME
20	Missing account number	MISSING_ACCOUNT_NUMBER
22	Unable to create account	ACCOUNT_CREATION_API_ERROR
23	Invalid contact	INVALID_CONTACT
24	Missing contact email	MISSING_CONTACT_EMAIL
25	Missing contact last name	MISSING_CONTACT_LAST_NAME
26	Unable to create contact	CONTACT_CREATION_API_ERROR
27	Multiple matching contacts found	MULTIPLE_CONTACTS_FOUND
28	Multiple matching accounts found	MULTIPLE_ACCOUNTS_FOUND
30	Invalid account owner	INVALID_ACCOUNT_OWNER
31	Invalid portal profile	INVALID_PORTAL_PROFILE

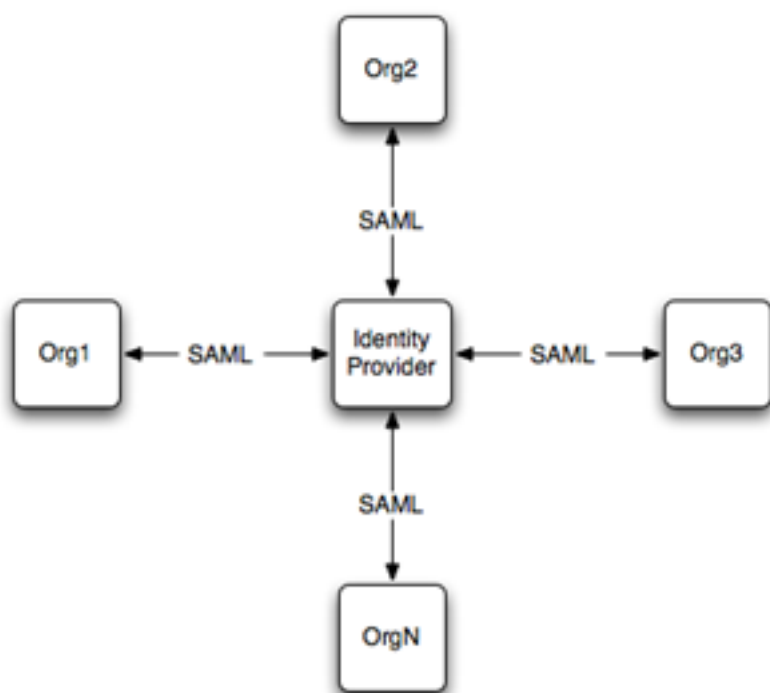
<b>Code</b>	<b>Description</b>	<b>Error Details</b>
32	Account change is not allowed	ACCOUNT_CHANGE_NOT_ALLOWED
33	Unable to update account	ACCOUNT_UPDATE_FAILED
34	Unable to update contact	CONTACT_UPDATE_FAILED
35	Invalid standard account field value	INVALID_STANDARD_ACCOUNT_FIELD_VALUE
36	Contact change not allowed	CONTACT_CHANGE_NOT_ALLOWED
37	Invalid portal role	INVALID_PORTAL_ROLE
38	Unable to update portal role	CANNOT_UPDATE_PORTAL_ROLE
39	Invalid SAML JIT Handler class	INVALID_JIT_HANDLER
40	Invalid execution user	INVALID_EXECUTION_USER
41	Execution error	APEX_EXECUTION_ERROR
42	Updating a contact with Person Account isn't supported	UNSUPPORTED_CONTACT_PERSONACCT_UPDATE

# CONFIGURE SSO ACROSS MULTIPLE SALESFORCE ORGS

Let your users log in across multiple Salesforce orgs using single sign-on (SSO) credentials. With SSO, you can validate user credentials against a corporate database or other app rather than managing separate passwords for each Salesforce org.

Enterprises often deploy more than one Salesforce org. Unless you implement SSO, users that access different orgs must reauthenticate with each org. Removing this extra login step makes it more convenient for users and enhances security because it's easier for users to maintain and use a single, strong password.

SSO follows a hub-and-spoke architecture. At the center is a centralized authentication hub, the identity provider. The identity provider validates credentials and asserts the user's identity to the spokes—Salesforce orgs that are the service providers. The org that is the identity provider generates SAML assertions that follow the SAML 2.0 standard for SSO.




Salesforce supports both identity provider–initiated and service provider–initiated logins.

## Set Up SSO from Salesforce to Salesforce

Configure one org as an identity provider and another org as a service provider.

1. [Enable and deploy a My Domain subdomain](#) in both Salesforce orgs.

From Setup, enter *My Domain* in the Quick Find box, and then select **My Domain**. Deploy the subdomain to the org's users.

 **Warning:** Deploying a domain on existing orgs can impact user bookmarks. Make sure that users are aware of this possibility before you deploy the subdomain on existing production orgs.


2. Set up one Salesforce org as the identity provider.

## Configure SSO Across Multiple Salesforce Orgs


- a. By default, Salesforce enables your org as an identity provider when you create the My Domain subdomain. To verify that your org is enabled as an identity provider, from Setup, enter *Identity Provider* in the Quick Find box, and select **Identity Provider**. If the org is not configured as an identity provider, click **Enable Identity Provider**.
    - b. Download a certificate from your identity provider. Your service provider uses a certificate to establish trust in the identity provider. If you previously haven't generated a certificate and key pair, when Salesforce enables an identity provider, it creates one. Optionally, you can pick an existing self-signed certificate or use a CA-signed certificate.
    - c. Copy the **Salesforce Identity** URL listed under SAML Metadata Discovery Endpoints on the Identity Provider page. You can use this URL in the next step to import SAML metadata when you configure SAML SSO settings on the service provider. Alternatively, to capture metadata in an XML file, click **Download Metadata**.
  3. Set up a service provider org.

Import metadata from a URL for the identity provider org. Alternatively, enter the SAML 2.0 settings manually, or import a metadata file with the settings.

    - a. From Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**.
    - b. Click **Edit**. To reveal the SAML SSO settings, select **SAML Enabled**. Save the change.
    - c. Under SAML Single Sign-On Settings, click the appropriate button to create a configuration.
      - **New**—Specify all settings manually.
      - **New from Metadata File**—Import SAML 2.0 settings from an XML file from your identity provider. This option reads the XML file and uses it to complete as many settings as possible.

 **Note:** If your XML file contains information for more than one configuration, the first configuration that occurs in the XML file is used.
      - **New from Metadata URL**—Import SAML 2.0 settings from a public URL listed under SAML Metadata Discovery Endpoints on the Identity Provider page. This option reads the XML file at this URL and uses it to complete as many settings as possible. The URL must be added to Remote Site Settings to access it from your Salesforce org.
    - d. Complete the [SAML settings that describe the identity provider](#). The metadata options populate most settings, making it easier to set up an org or community as a service provider. For example, the metadata populates the Issuer, Entity ID, and Identity Provider Login URL fields automatically with URLs for your identity provider.

To manage SSO effectively across multiple Salesforce orgs, specify federation ID as the SAML identity type.

 **Note:** The structure of a Salesforce username is unique to each Salesforce org. Specifying federation ID as the SAML identity type allows an identity provider to map user identities across orgs. Each user has a common federation ID but a unique username across orgs.

After completing the settings, save them. From this page, copy the values from the Entity ID and Identity Provider Login URL fields. You need these values later when you define a connected app on the identity provider.
    - e. On the service provider, add the identity provider as an authentication service. From Setup, enter *My Domain* in the Quick Find box, then select **My Domain**. Under Authentication Configuration, click **Edit** and select the authentication service for your identity provider. Save the settings.
    - f. To define other orgs as service providers, repeat these steps.
  4. On the identity provider org, [create a Salesforce connected app](#).

Creating a connected app defines the service provider to the identity provider org.

    - a. Use the New Connected App wizard to define a connected app.

## Configure SSO Across Multiple Salesforce Orgs

- In Lightning Experience, from Setup, enter *App* in the Quick Find box, then select **App Manager**. Click **New Connected App**.
  - In Salesforce Classic, from Setup, enter *Apps* in the Quick Find box, then select **Apps**. On the page under Connected Apps, click **New**.
- Remember the Entity ID and Login URL values that you saved? You need them now to configure settings for the connected app. Under Web App Settings, click **Enable SAML**. For **Entity ID**, paste in the Entity ID from the SAML SSO settings. For the **ACS URL**, enter the saved endpoint Login URL (for the org or community that's your service provider). To map user identities across multiple Salesforce orgs, specify **Federation ID** for the subject type. Save the settings.
  - To select the profiles and permission sets allowed to access this service provider, click **Manage** on the connected app page. Select **Manage Profiles** or **Manage Permission Sets**. Add the profiles or permission sets for users who can access this app.
  - When complete, Salesforce lists the new service provider on your identity provider's list of connected apps.
  - To define connected apps for other service provider orgs or communities, repeat these steps as needed.  
To learn how to set up a Salesforce mobile app as a connected app, see this [Trailhead](#).
5. Test the SSO implementation.
- Create a test user in your identity provider org, and set the user's federation ID to a unique value. Make sure that you assign the user a profile that has been granted access to your service provider.
  - Create a test user in your service provider org, and set the user's federation ID to the same value as your test user in the identity provider. This action binds the two accounts together.
  - Log out from both orgs.
  - Enter the My Domain URL of the identity provider org into your browser, for example, `https://idp.mydomain.salesforce.com/`.
  - Log in to your identity provider org as the test user.
  - Now, enter the URL of the service provider org into your browser, for example, `https://sp1.mydomain.salesforce.com/`.  
You are redirected to the identity provider org. Because you're already authenticated, you are redirected back to your service provider org. Presto, you're logged in!

## Configure SSO from an Org to a Community

---

Implementing SSO between an org and a community is much the same as configuring SSO between two Salesforce orgs. What's different is that you must specify endpoint URLs that point to the community.

- To set up a community as a service provider, use the community URL under SAML Metadata Discovery Endpoints on the Identity Provider page. Upload the SAML metadata from this URL. Using the metadata populates the service provider's SAML SSO settings, including the Login URL that points to the community. When you define a connected app on the identity provider, specify this Login URL as the ACS URL.
- To set up a community as an identity provider, complete the Identity Provider Login URL on the SAML Single Sign-On Settings page in the service provider org. This URL is where Salesforce sends a SAML request to start a login sequence. Use the URL for the community as the Identity Provider Login URL, rather than the URL for the org's My Domain subdomain. For example, to set up a community as an identity provider, specify the Identity Provider Login URL with the HTTP redirect binding.

`https://acme.force.com/customers/idp/endpoint/HttpRedirect`

In contrast, to set up an org as an identity provider, use the org's My Domain subdomain.

`https://acme.my.salesforce.com/idp/endpoint/HttpRedirect`

#### SEE ALSO:

[Salesforce Help: Identity Providers and Service Providers](#)

[Configuring SAML SSO for a Canvas App](#)

[Configuring SSO for Mobile and Desktop Apps Using SAML and OAuth](#)

## Best Practices and Tips for Implementing Single Sign-On

Salesforce offers a set of best practices that you can follow when implementing delegated authentication, federated authentication using SAML, single sign-on (SSO) for portals, and SSO for Sites.

Salesforce offers the following ways to use SSO.

- Federated authentication using Security Assertion Markup Language (SAML) lets you send authentication and authorization data between affiliated but unrelated web services. You can log in to Salesforce from a client app. Salesforce enables federated authentication for your org automatically.
- Delegated authentication SSO integrates Salesforce with an authentication method that you choose. You can integrate authentication with your LDAP (Lightweight Directory Access Protocol) server or use a token instead of a password for authentication. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Delegated authentication offers the following benefits.

- Uses a stronger form of user authentication, such as integration with a secure identity provider
- Makes your login page private and accessible only behind a corporate firewall
- Differentiates your org from all other companies that use Salesforce to reduce phishing attacks

You must contact Salesforce to enable delegated authentication before you can configure it on your org.

- Authentication providers let your users log in to your Salesforce org using their login credentials from an external service provider. Salesforce supports the OpenID Connect protocol, which lets users log in from any OpenID Connect provider, such as Google, PayPal, and LinkedIn. When an authentication provider is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

In addition, you can also configure SAML for use with portals as well as for Sites.

## Delegated Authentication Best Practices

Consider these best practices when implementing delegated authentication SSO for your org.

- Your org's implementation of the web service must be accessible by Salesforce servers, so you must deploy the web service on a server in your DMZ. Remember to use your server's external DNS name when entering the delegated gateway URL in the Delegated

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Customer Portals and partner portals are not available in **Database.com**

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application

AND

Modify All Data



authentication section in Salesforce. From Setup, enter *Single Sign-On Settings* in the *Quick Find* box, then select **Single Sign-On Settings**.

- If Salesforce and your system can't connect, or if the request takes longer than 10 seconds to process, the login attempt fails. The user gets an error message indicating that the corporate authentication service is down.
- Namespaces, element names, and capitalization must be exact in SOAP requests. Wherever possible, generate your server stub from the WSDL file to ensure accuracy.
- For security reasons, make your web service available by TLS. A certificate from a trusted provider, such as Verisign or Thawte, is required. For a list of trusted providers, contact Salesforce.
- The IP address that originated the login request is *sourceIp*. Use this information to restrict access based on the user's location. Also, the Salesforce feature that validates login IP ranges applies to SSO users. For more information, see [Restrict Where and When Users Can Log In to Salesforce](#).
- You might need to map your org's internal usernames to your Salesforce usernames. If your org doesn't follow a standard mapping, try extending your user database schema (for example, Active Directory) to include the Salesforce username as an attribute of a user account. Your authentication service can then use this attribute to map back to a user account.
- We recommend that you don't enable SSO for Salesforce admins. If your Salesforce admins are SSO users and your SSO server has an outage, they have no way to log in to Salesforce. Make sure that Salesforce admins can log in to Salesforce so that they can disable SSO if problems occur.
- We recommend that you use a Developer Edition account or a sandbox when developing a SSO solution before implementing it in your org. To sign up for a free Developer Edition account, go to [developer.salesforce.com](https://developer.salesforce.com).
- Make sure to test your implementation with Salesforce clients, such as Salesforce for Outlook, Connect for Office, and Connect Offline. For more information, see [Single Sign-On for Salesforce clients](#).

## Federated Authentication Using SAML Best Practices

Consider these best practices when implementing federated SSO with SAML for your org.

- Get the Salesforce login URL from the Single Sign On Settings configuration page and enter it in the corresponding configuration parameter of your identity provider. Sometimes, the setting is called the recipient URL.
- Salesforce allows a maximum of 3 minutes for clock skew with your IDP server. Make sure that your server's clock is up to date.
- If you can't log in with SAML assertion, check the login history and note the error message. Use the SAML Assertion Validator on the Single Sign On Settings configuration page to troubleshoot.
- Map your orgs internal usernames and Salesforce usernames. To map the names, you can add a unique identifier to the `FederationIdentifier` field of each Salesforce user. Or you can extend your user database schema (for example, Active Directory) to include the Salesforce username as an attribute of a user account. Choose the corresponding option for the `SAML Identity Type` field, and configure your authentication service to send the identifier in SAML assertions.
- Before allowing users to log in with SAML assertions, enable the SAML org preference and provide the necessary configurations.
- Use the My Domain feature to prevent users from logging in to Salesforce directly, and give admins more control over login policies. You can use the URL parameters provided in the `Salesforce Login URL` value from the Single Sign-On Settings configuration page with your custom domain.

For example, if the `Salesforce Login URL` is `https://login.salesforce.com/?saml=02HKiP...`

you can use `https://yourDomain.my.salesforce.com/?saml=02HKiP...`

- We recommend that you use a Developer Edition account or a sandbox when testing a SAML SSO solution. To sign up for a free Developer Edition account, go to [developer.salesforce.com](https://developer.salesforce.com).
- Sandbox copies are made with federated authentication with SAML disabled. Any configuration information is preserved, except the value for `Salesforce Login URL`. The `Salesforce Login URL` is updated to match your sandbox URL, for

example `https://yourInstance.salesforce.com/`, after you re-enable SAML. To enable SAML in the sandbox, from Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**; then click **Edit**, and select **SAML Enabled**.

- Your identity provider must allow you to set the service provider's audience URL. The value must match the `Entity ID` value in the SSO configuration. The default is `https://saml.salesforce.com`.

## SSO for Portals Best Practices

Customer Portals and partner portals are not available for new orgs as of the Summer '13 release. Use Communities instead. For more information about SSO and SAML for Communities, see "Configuring SAML for Communities" in the Salesforce Help. If you continue to use portals, be aware of these requirements.

- Only SAML version 2.0 can be used with portals.
- Only Customer Portals and partner portals are supported.
- Service provider initiated login is not supported.
- Both the `portal_id` and `organization_id` attributes are required. If only one is specified, the user receives an error.
- If both the `portal_id` and `organization_id` attributes are populated in the SAML assertion, the user is directed to that portal login. If neither is populated, the user is directed to the regular SAML Salesforce login.
- More than one portal can be used with a single org.

## SSO for Sites Best Practices

- Only SAML version 2.0 can be used with Sites.
- Only Customer Portals and partner portals are supported.
- Service provider initiated login is not supported.
- The `portal_id`, `organization_id`, and `siteUrl` attributes are required. If only one is specified, the user receives an error.
- If all the `portal_id`, `organization_id` and `siteUrl` attributes are populated in the SAML assertion, the user is directed to that Sites login. If the `siteUrl` isn't populated and the other two are, the user is directed to the portal login.
- More than one portal can be used with a single org.

## SSO Login Settings Tips

- You can set a user permission to prevent users from using a Salesforce username and password. For example, use this permission when you configure users to use an authentication provider for single sign-on, and want them to use that authentication provider, only. Assign these users, or the profile for these users, the "Is Single Sign-On Enabled" user permission. If the "Is Single Sign-On Enabled" permission is not available in your org, contact Salesforce and ask Support to enable the delegated authentication feature. In this case, you don't have to configure delegated authentication for your org. However, you need the delegated authentication feature to enable the "Is Single Sign-On Enabled" permission for users or profiles.
- System administrators should always be able to log in to Salesforce, even if single sign-on is enabled for their accounts. For example, if your third-party authentication provider has an outage, the administrators need a way to log in to Salesforce. And, if an authentication provider has an outage, the system administrators may configure other users to log in to Salesforce.

## Configuring SSO for Mobile and Desktop Apps Using SAML and OAuth

Salesforce mobile and desktop clients, including the SalesforceA mobile app for administrators, can combine OAuth and SAML protocols for service provider–initiated single sign-on (SSO).

### App Support for SSO

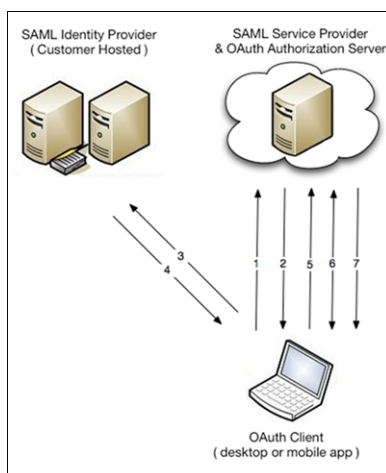
To authenticate mobile and desktop clients, a Salesforce org configured as a service provider can combine the OAuth and SAML protocols. OAuth allows users to connect applications to their accounts. SAML authenticates those connections. Using OAuth and SAML, mobile and desktop clients can take advantage of SSO integration in a way similar to web applications.

SSO integration is based on several core tenets.

- Developers are increasingly rewriting desktop and mobile applications to use OAuth to connect to user accounts. At runtime, users authenticate and authorize the app. After this initial step, a high-entropy (long, random) token is issued to the device. It is used instead of a password the next time the application is invoked. The token is unique to the user and application combination, and it is independently monitored, managed, and revoked.
- Many applications are now using web browsers to authenticate instead of native code. By using browsers, these applications can take advantage of SSO protocols written for the web. Previously, most applications were hard coded to ask users for credentials.
- Applications are separating authentication from authorization. By uncoupling these functions, an application adapts to change more easily. For instance, a client that normally prompts a user to log in using a web page from its servers can instead use SAML. It's also easy to implement adaptive risk-based authentication techniques. These measures remove the need for other, more cumbersome security measures, such as a Salesforce API token, which is a second credential that some applications use.
- Deployment must be simple and standard. Organizations that deploy SSO want to do so once and have it work everywhere.

### Combining SAML and OAuth

By layering the SAML and OAuth protocols, mobile and desktop clients perform SSO using the process shown.



1. The OAuth client makes an authorization request to the hostname you specify. Using an embedded browser, the client asks the service provider for authorization. It does so using a custom URL that is your My Domain subdomain.
2. The authorization server detects that the client must authenticate and redirects the user to the SAML identity provider (IdP). The URL for the authorization server is passed via the RelayState parameter.
3. The user accesses the IdP, and the IdP performs authentication.

4. After the user is authenticated, the IdP sends back a SAML response. The browser transmits a response with a RelayState parameter. The response indicates that the client app is returning to the OAuth authorization server.
5. Salesforce processes the SAML assertion and logs the user in. The digital signature applied to the SAML response verifies that the message is from your system. At this point, Salesforce authenticates the user and redirects them to the authorization server.
6. After authentication, the client prompts the user to allow the client to connect to their account. The prompt is a simple web page that shows the user information about the client and what it's requesting.
7. If the user approves the application, it is issued a high-entropy token that the application uses to establish a session. Subsequent application use does not require the user to reenter credentials.

When layered with SAML, OAuth is much like any other bookmark or deep link. No additional development or deployment steps are necessary to enable SSO for the client app.

## Configure Service Provider–Initiated SSO

Mobile or desktop apps that layer the SAML and OAuth protocols require two configuration steps.

1. Use the My Domain wizard to [set up a subdomain](#). Deploying a subdomain improves the user experience and allows users to access deep links. This step is an easy way to prepare your org for service provider–initiated SSO from a mobile app or desktop client.
2. Point the client at the host that your subdomain represents. This step involves defining a new host connection in the app's settings.

Let's look at some examples that configure mobile and desktop clients for service provider–initiated SSO.

- [Salesforce for iOS or Android](#) on page 48
- [SalesforceA for iOS](#) on page 49
- [SalesforceA for Android](#) on page 50
- [Desktop clients](#) on page 50

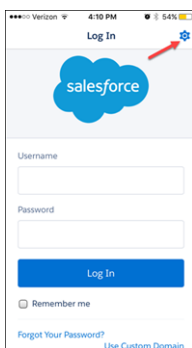
Various Salesforce and ISV applications support SSO using layered SAML and OAuth protocols. Because the combined protocol approach relies on open standards and public APIs, you can also use it for custom app development. If you are a developer that wants to take this approach and follow best practices, see [Advice for Application Developers](#) on page 51.

## Salesforce for iOS or Android

The Salesforce mobile app works only with a Salesforce org configured for service provider–initiated SSO.

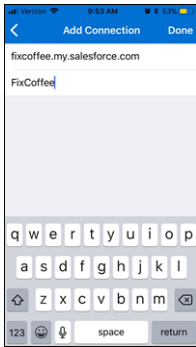
To set up Salesforce on either an iOS or Android device, configure your My Domain URL under the app's connection settings.

1. For example, on iOS, launch the app. At the top right of the Log In page, tap the gear icon.

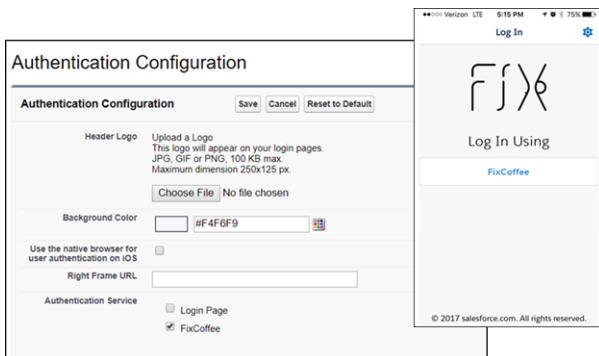


2. To add a login connection, tap the + icon.

3. Enter the My Domain subdomain for your Salesforce org. Don't include `https://` in the URL. To save the new connection, tap **Done**.



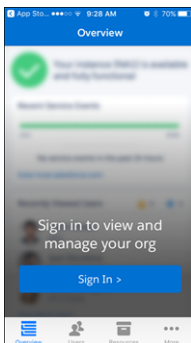
4. Select the connection you created, which redirects you to your org.
5. The login page appears. The login options depend on how you configured authentication services on your My Domain page. In this example, the login page is customized. The standard login authentication service is disabled, and the FixCoffee authentication service for the sample org is activated.



6. Enter your credentials, and log in to Salesforce.

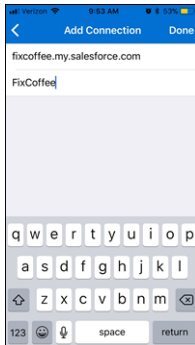
## SalesforceA for iOS

1. Launch the application, and tap Sign In.



2. Configure your My Domain subdomain as a new connection under the app's connection settings. To add a connection, tap the gear icon in the top right, and then the + icon.

3. Enter the My Domain subdomain for your Salesforce org. Don't include `https://` in the URL. To save the new connection, tap **Done**.



## SalesforceA for Android

To set up SalesforceA on an Android device, configure a new connection that points to your subdomain.

1. Launch the SalesforceA app, and access the menu.
2. Tap **Change Server**, and then tap **Add Connection**.
3. Enter the My Domain subdomain, including `https://`. To apply the URL to the connection, tap **Apply**.
4. Choose the new connection, and tap **Apply**.

## Desktop Clients

Several desktop clients, including Chatter Desktop, Salesforce for Outlook, and Data Loader, also layer the SAML and OAuth protocols. It's easy to configure these desktop apps for service provider–initiated SSO using a similar process.

1. Enable and deploy a My Domain subdomain in your Salesforce org.
2. Configure a new connection in the desktop client that points to your My Domain subdomain.

For example, to configure Data Loader as a desktop client, click **Use Custom Domain** on the login page. Enter the My Domain subdomain.

## Advice for Deploying Applications

When deploying SSO for devices, consider the following best practices.

- Confirm that service provider–initiated SSO is working properly. Verify SSO using a desktop browser before trying it on a mobile device. Some deployments have difficulty properly propagating the RelayState parameter through the SAML request and response sequence. Verify that the IdP endpoints are correct, the RelayState's URL encoding is maintained, and the value sent to the IdP is echoed back to Salesforce.

 **Note:** The returned value must exactly match what's sent.

- Clearly communicate your My Domain URL to users and provide client-specific instructions. Users bear the responsibility of properly configuring apps to point to the right URL. Make sure that you educate users on the proper configuration steps.
- Consider the impact of IP restrictions on mobile devices. VPN and BES servers often help to alleviate issues. To circumvent IP restrictions, some mobile clients send activation emails.
- Assess the design of your identity provider's login page. Both the size and performance of mobile client pages can impact a user's experience. While Salesforce login services can dynamically adapt to various devices, when hosting your own identity provider,

consider the size and loading speed of your login page. Consider implementing user agents that detect less-capable devices, and tailor or simplify authentication interfaces. This precaution is especially important when authenticating users on older device types.

## Advice for Application Developers


When building OAuth-enabled applications, consider these best practices.

- Allow users to specify their My Domain URL. When deploying an application, it's important that users can specify the login service for authentication. Best practice is to allow a user to choose between production, sandbox (or test), and custom orgs. To use SSO, users must be able to specify a custom host connection. Also, consider techniques that simplify configuration, such as allowing an enterprise IT group to centrally manage user configurations.
- Consider the size of the login window. Given the wide variety of identity providers, it is difficult to predict the look and feel of the page presented when users log in using SAML. While some platforms, such as Android and iOS, can gracefully adapt to different page sizes, other platforms are less flexible. In these cases, consider using a reasonably large authentication interface, or allow the user to resize or scroll the interface.

## Delegated Authentication

Another option to authenticate users on mobile and desktop devices is delegated authentication. If enabled for an org and in user profiles, when a user authenticates directly to Salesforce, credentials are sent back to a customized endpoint over HTTPS.

Salesforce uses this process for delegated authentication.

1. When a user tries to log in, Salesforce validates the username and checks the user's profile settings.
2. If the profile has the Is Single Sign-On Enabled user permission, Salesforce does not validate the username and password. Instead, a web service call is made to the org, asking it to validate the username and password.  
 **Note:** Salesforce doesn't store, log, or view the password in any way. It is disposed of immediately after the process is complete.
3. The web service call passes the username, password, and IP address to a web service that you host.
4. Your implementation of the web service validates the information and returns either true or false. If the response is true, the login process continues, a new session is generated, and the user proceeds to the application. If the web service returns false, the user is informed that the username and password combination is invalid.

This process allows existing authentication systems to validate credentials, and it works with all mobile and desktop clients. After a delegated authentication service is built and enabled for a user, no additional configuration is required. The user can log in using the regular Salesforce login page, but your web service validates credentials.

## Adapt Existing Delegated Authentication-Based SSO

Historically, many Salesforce customers have implemented SSO on top of delegated authentication, which follows this process.

1. Users start at an SSO page, perhaps on their corporate intranet.
2. Users log in locally to their own authentication system.
3. The SSO page generates a cryptographic token.
4. The Salesforce username and token are posted to the Salesforce login page. In addition, a startURL parameter is often included to indicate the user's starting page.
5. If the user profile has the Uses Single Sign-on permission enabled, Salesforce makes a web service call to a custom SSO service. Salesforce asks the service to validate the username and token.

6. The service validates the cryptographic token and returns either true or false. If the response is true, the login process continues, a new session is generated, and the user proceeds to the application.

SAML, an industry standard, has superseded [delegated authentication](#). By using a SAML approach, companies can establish effective and standards-based SSO implementations.

If you have an existing SSO deployment using delegated authentication, you can apply the following technique with mobile or desktop clients. While delegated authentication deployments do not use SAML, they can be adapted to take advantage of these capabilities by using part of the SAML protocol. This change requires small adjustments to the code in your SSO service.

1. Configure My Domain and SAML for your org. You can configure dummy values for the SAML settings for everything except the Identity Provider Login URL. This setting must be a URL that receives the SAML authentication request from Salesforce, but it doesn't need to actually process the request.
2. A user requests a resource in the org, for example, `https://customer.my.salesforce.com/001/o`.
3. Salesforce notices that the user does not have a session for that org and sends a SAML authentication request to the org's SSO service. The request includes the RelayState parameter, for example, `/001/o`.
4. The SSO service receives both the SAMLRequest and RelayState parameters via an HTTP POST operation. The service ignores the SAMLRequest, but picks up the RelayState.
5. The SSO service authenticates the user and generates a cryptographic token.
6. The Salesforce username and token are posted to the Salesforce login page. In addition, a startURL parameter is echoed back with the value from the RelayState parameter.
7. If the user's profile has the Uses Single Sign-on permission enabled, Salesforce makes a web service call to the SSO service, asking it to validate the username and token.
8. The service validates the cryptographic token and returns either true or false. If the response is true, the login process continues, a new session is generated, and the user proceeds to the requested page.

To summarize, Salesforce sends the user request over SAML, but the request is sent back using delegated authentication. The RelayState parameter is transformed into the startURL parameter to redirect the user to the correct page. Using this technique, mobile and desktop clients that use SAML with Salesforce can also take advantage of SSO over delegated authentication.

SEE ALSO:

[Configure SSO Across Multiple Salesforce Orgs](#)

[Salesforce Help: Configure SSO to Salesforce Using Microsoft Active Directory Federation Services as the Identity Provider](#)

[Salesforce Help: Salesforce for Outlook](#)

[Trailhead: Use the Salesforce Mobile App with Single Sign-On](#)

## Configuring SAML SSO for a Canvas App

---

Configuring SAML single sign-on (SSO) for a canvas app lets users easily access a new or existing application as a part of their Salesforce experience. A canvas app in one Salesforce org functions as an identity provider, authenticating another Salesforce org that's the service provider. The canvas app, hosted in the identity provider org, uses a signed request to reference a Visualforce page in the second org.

### Create and Deploy My Domain Subdomain

In each org, use the Salesforce My Domain wizard to [deploy a subdomain](#) under `my.salesforce.com`.


If you haven't previously generated a certificate and key pair, setting up a My Domain subdomain also creates them. In a later step, you can use this certificate, or a CA-signed certificate or other self-signed certificate, to establish trust between your Salesforce orgs.



To provide information about your identity provider to your service provider, download the identity provider metadata.


1. From Setup, enter *Identity Provider* in the Quick Find box, and select **Identity Provider**.
2. Click **Download Metadata**. The metadata includes URLs and a self-signed certificate that you can use in a later step. If you want to copy the certificate from a separate file, click **Download Certificate**.

On the Identity Provider page, under SAML Metadata Discovery Endpoints, copy the Salesforce Identity URL. You can use this URL to import SAML metadata when you configure SAML SSO settings on the service provider.

 **Warning:** Deploying a domain on existing orgs can impact user bookmarks. Make sure that your users are aware of this possibility before you deploy the subdomain on existing production orgs.


## Configure SAML Settings on the Service Provider

1. From Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**.
2. Click **Edit**. To reveal the SAML SSO settings, select **SAML Enabled**. Save your changes.
3. Under SAML Single Sign-On Settings, click one of the buttons to create a configuration.
  - **New**—All settings are manually specified.
  - **New from Metadata File**—Imports SAML 2.0 settings from an XML file from your identity provider. This option uses the XML file to complete as many settings as possible.

 **Note:** If your XML file contains information for more than one configuration, the first configuration that occurs in the XML file is used.

- **New from Metadata URL**—Imports SAML 2.0 settings from the public URL listed under SAML Metadata Discovery Endpoints on the Identity Provider page. This option uses the XML file to complete as many settings as possible. To access the URL from your Salesforce org, it must be added to Remote Site Settings.

Complete the [SAML settings that describe the identity provider](#). When using metadata, most settings are automatically populated, making it easier to set up an org or community as a service provider. For example, the metadata populates the Issuer, Entity ID, and Identity Provider Login URL fields with URLs for your identity provider.

 **Note:** The structure of a Salesforce username is unique to each Salesforce org. To manage SSO across multiple Salesforce orgs, select **Federation ID** as the SAML identity type. This setting allows an identity provider to map user identities across orgs. Each user has a common federation ID across multiple orgs, but a unique username in each org.

**SAML Single Sign-On Setting**

**SAML Single Sign-On Setting Edit** Save Save & New Cancel

Name MY\_SAML API Name MY\_SAML

SAML Version 2.0 User Provisioning Enabled

Issuer https://idp-dev-ed.blt202 Entity ID https://sp-dev-ed.blt202

Identity Provider Certificate Browse SelfSignedCert\_15Apr2014\_224531.crt

Signing Certificate Default Certificate

Assertion Decryption Certificate Assertion not encrypted

SAML Identity Type
 

- Assertion contains User's salesforce.com username
- Assertion contains the Federation ID from the User object
- Assertion contains the User ID from the User object

SAML Identity Location
 

- Identity is in the NameIdentifier element of the Subject statement
- Identity is in an Attribute element

Identity Provider Login URL https://idp-dev-ed.blt202.salesforce.com/idp/endpoint?httpPost

Identity Provider Logout URL

Custom Error URL

Service Provider Initiated Request Binding
 

- HTTP POST
- HTTP Redirect

Save Save & New Cancel


4. Save your SAML settings.
5. Copy and save the Entity ID and the Login URL for the service provider org listed under Endpoints. You need these values when defining a connected app for the service provider.

6. Add the identity provider as an authentication service.
  - a. From Setup, enter *My Domain* in the Quick Find box, then select **My Domain**.
  - b. Under Authentication Configuration, click **Edit** and check the authentication service for your identity provider.
  - c. Save the settings.

## Create a Connected App in the Identity Provider

Creating a connected app defines the service provider to the identity provider.

1. Log in to the Salesforce org that's the identity provider.
2. Use the connected app wizard to define a connected app.
  - In Lightning Experience, from Setup, enter *App* in the Quick Find box, and select **App Manager**. Click **New Connected App**.
  - In Salesforce Classic, from Setup, enter *Apps* in the Quick Find box, and select **Apps**. Under Connected Apps, click **New**.
3. Under Web App Settings, click **Enable SAML** to enter the Entity ID and Login URL values that you saved.
  - a. For Entity ID, enter the value from the SAML SSO settings.
  - b. For ACS URL, enter the saved endpoint login URL of the org or community that's your service provider.
  - c. To map user identities across multiple Salesforce orgs, select **Federation ID** for Subject Type. A federation ID is a unique value assigned to the user that lets you send authentication and authorization data between affiliated but unrelated web services.
  - d. Save the settings.

 **Note:** It can take a few minutes for Salesforce to create the connected app.
4. To select the profiles and permission sets allowed to access this service provider, click **Manage** on the connected app page. Select **Manage Profiles** or **Manage Permission Sets**. Add the profiles or permission sets for users who can access this app.
 

When complete, Salesforce lists the new service provider in your identity provider's list of connected apps.

**New Connected App**

Save Cancel

To publish an app, you need to have chosen a namespace prefix. [Click here to choose a namespace prefix.](#)

**Basic Information**

Connected App Name Burlington Service Provider

API Name Burlington\_Service\_Provider

Contact Email dsiebold@salesforce.com

Contact Phone

Logo Image URL Upload logo image or Choose one of our sample logos

Icon URL Choose one of our sample logos

Info URL

Description

**API (Enable OAuth Settings)**

Enable OAuth Settings

**Web App Settings**

Start URL

Enable SAML

Entity Id https://sp-dev-ed3184023111.salesforce.com

ACS URL https://sp-dev-ed3184023111.salesforce.com?so=00DD00000077Lsr

Subject Type Federation ID

Name ID Format urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

Issuer https://idp-dev-ed3184023111.salesforce.com

Verify Request Signatures

Encrypt SAML Response

## Test the Connected App

Be sure that you're not logged in to any other SSO apps when you test your configuration. Otherwise, authentication fails, and you get an error that authentication can't proceed using the identity provider certificate.

1. Log in to your Salesforce org that's the identity provider.
2. Create a test user in your identity provider org, and set the user's federation ID to a unique value. Make sure that you assign the user a profile that was granted access to your service provider.
3. Create a test user in your service provider org. Set the user's federation ID to the same value as your test user in the identity provider org. This action binds the two accounts together.
4. Log out from both orgs.
5. In a browser, enter the My Domain URL of the identity provider org, for example, `https://idp.my.salesforce.com/`.
6. Log in to your identity provider org as the test user.
7. In a browser, enter the URL of the service provider org, for example, `https://sp.my.salesforce.com/`.

You are redirected to the identity provider org. Because you're already authenticated, you are redirected back to your service provider org and logged in.

## Create an Identity User

Set up your canvas app, and configure it to use SSO. Create an identity user in the Salesforce identity provider org and then bind that user to the admin user of the service provider org.

1. Log in to your org that's the identity provider.
2. Create a user as an identity user. Be sure to enter a unique value for the federation ID.

3. Log in to your service provider org. To bind the identity user to the service provider administrator, set the federation ID for the administrator to the same value as the ID of the identity user.
4. Log out of both orgs.
5. In a browser, enter the domain URL of the identity provider and log in as the identity user.
6. In the browser, enter the domain URL of the service provider. A button appears on the login page with the name of your connected app. Click the button.

The browser redirects to the identity provider and then back to the service provider. You are logged in to the service provider as the administrator because the federation ID binds the accounts together.

## Create a Visualforce Page in the Service Provider

1. Log in as the administrator to your org that's the service provider.
2. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
3. Click **New**.
4. For Label, enter a label for your Visualforce page. For example, enter *ServiceProviderPage*.
5. Replace the code in the Visualforce Markup tab with this code sample.

```

1 <apex:page showHeader="false">
2
3 <h1>Congratulations</h1>
4
5 My service provider Visualforce page
6
7 </apex:page>

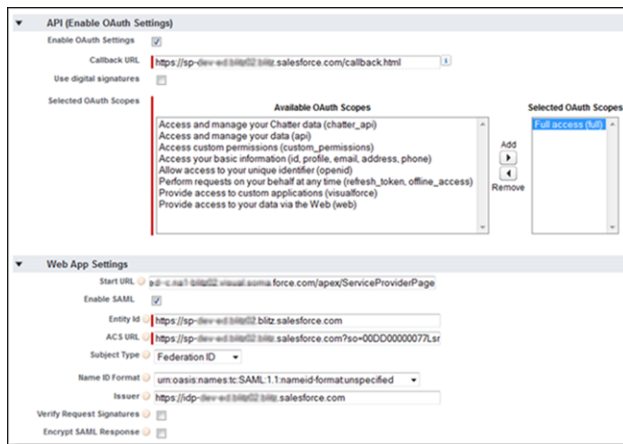
```

6. Save the code.
7. Click **Preview** to review the Visualforce page.
8. Save the page URL, for example, *https://sp-`<instance_name>`.force.com/apex/ServiceProviderPage*.
9. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
10. To give users access to the Visualforce page, click **Security** next to the page name. Add the Standard User to the list of enabled profiles for this page, and save the settings.

## Update the Connected App in the Identity Provider

1. Log in to your Salesforce org that's the identity provider.
2. Modify the settings for the connected app defined previously.
  - In Lightning Experience, from Setup, enter *App* in the Quick Find box, and select **Manage Connected Apps**.
  - In Salesforce Classic, from Setup, enter *Apps* in the Quick Find box, and select **Apps**.
3. Next to your connected app, click **Edit**.
  - a. For the Start URL, enter the URL of the Visualforce page you created in the service provider.
  - b. Under API (Enable OAuth Settings):
    - Select **Enable OAuth Settings**.


- For Callback URL, enter any URL. For example, `https://sp-<instance_name>.salesforce.com/callback.html`. You can use any value because you are using signed request authentication.
  - For Selected OAuth Scopes, select **Full Access (full)**.
- c. Under Canvas App Settings:
- Select **Force.com Canvas**. This setting specifies that the connected app is a canvas app.
  - For Canvas App URL, enter the same URL that you entered for the Start URL.
  - For Access Method, select **Signed Request (POST)**.
  - For SAML Initiation Method, select **Identity Provider Initiated**. This setting indicates that the identity provider makes the initial request to start the SSO flow.
  - For Locations, add the Chatter tab to the Selected list of locations.
- d. Save the settings.



#### 4. (Optional) Remove the login page from the service provider org.

If you selected **Service Provider Initiated** as the SAML Initiation Method when defining canvas app settings for your connected app, remove the login page from your service provider org. Doing so allows the service provider org to call the SAML SSO flow directly when it invokes the canvas app. In addition, the login page can't be contained in an iframe.

If you selected **Identity Provider Initiated** instead, you can keep the login page active for your service provider org.

 **Note:** Log in to the service provider org as an administrator in a different browser. This way, if you run into issues, you can enable the login page.

- To remove the login page from the service provider org, log in as the administrator to the org.
- From Setup, enter *My Domain* in the Quick Find box, then select **My Domain**.
- Under Authentication Configuration, click **Edit**.
- Deselect **Login Page** and save the settings.

## Test the Canvas App

- Log in as a standard user to your org that's the identity provider.

2. Click the **Chatter** tab. You see a link to the canvas app. The canvas app is actually a Visualforce page that resides in the service provider org.
3. Click the canvas app link. If the Visualforce page appears, the identity provider org authenticated the user into the service provider org.

## Add Code to the Visualforce Page to Retrieve the Signed Request

When your canvas app appears, you know that SAML SSO authentication is working. However, as a result of the redirect, the initial signed request is no longer available. The signed request contains the context information and the required authentication token to communicate with the identity org. By adding some code to your Visualforce page, you can retrieve the signed request.

1. Log in as the administrator to your org that's the service provider. If you removed the login page, log in to the identity provider org as the identity user, and replace the browser URL with the service provider URL.
2. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
3. Next to the name of your Visualforce page, click **Edit**.
4. Replace the code in the Visualforce Markup tab with this code.

```

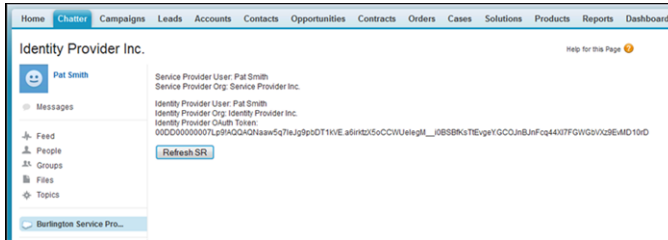
01 <apex:page showHeader="false">
02 <script type='text/javascript' src='/canvas/sdk/js/canvas-all.js' />
03 <script>
04     function refreshSR() {
05         Sfdc.canvas.client.refreshSignedRequest(function(data) {
06             if (data.status === 200) {
07                 var signedRequest = data.payload.response;
08                 var part = signedRequest.split('.')[1];
09                 var obj = JSON.parse(Sfdc.canvas.decode(part));
10                 updateDisplay(obj);
11             }
12         });
13     }
14     function updateDisplay(obj) {
15         var oauth = document.getElementById('oauth');
16         oauth.innerHTML = obj.client.oauthToken;
17         var user = document.getElementById('user');
18         user.innerHTML = obj.context.user.fullName;
19         var org = document.getElementById('org');
20         org.innerHTML = obj.context.organization.name;
21     }
22 </script>
23 <p/>
24 Service Provider User: {!$User.FirstName} {!$User.LastName}<br/>
25 Service Provider Org: {!$Organization.Name}<br/>
26 <p/>
27 Identity Provider User: <span id="user"></span><br/>
28 Identity Provider Org: <span id="org"></span><br/>
29 Identity Provider OAuth Token: <span id="oauth"></span><br/>
30 <p/>
31 <input id="refresh" type="button" value="Refresh SR" onclick="refreshSR();" />
32 </apex:page>

```

5. Save the code.
6. Log out of both orgs.

## Test the Updated Page

1. Log in as a standard user to the identity provider.
2. Click the **Chatter** tab. A link to the canvas app appears.



3. Click the link to the canvas app, and the Visualforce page appears. When you click **Refresh SR**, the page retrieves the signed request information for the user in the current org, which is the identity provider org.

You did it! You configured SAML SSO for a canvas app. Now you can give users seamless and secure access to your canvas apps.

### SEE ALSO:

[Salesforce Help: Identity Providers and Service Providers](#)

[Canvas Developer Guide](#)

[Configure SSO Across Multiple Salesforce Orgs](#)

# ENABLE SINGLE SIGN-ON FOR PORTALS


Single sign-on (SSO) lets users access authorized network resources with one login. You validate usernames and passwords against your corporate user database or other client app rather than Salesforce managing separate passwords for each resource.

You can set up Customer Portals and partner portals to use [SAML single sign-on](#), so that a customer only has to login once.

 **Note:** Single sign-on with portals is only supported for SAML 2.0.

To enable single sign-on for portals:

1. In addition to the [SAML sign-on information](#) that must be gathered and shared with your identity provider, you must supply your information provider with the Organization ID and the Portal ID. In the SAML assertion that is sent from your identity provider, the `portal_id` and `organization_id` must be added as attributes.

 **Note:** You can leave these attributes blank to differentiate between portal and platform users. For example, when blank, the user is a regular platform user and when populated, the user is a portal user.

- a. From Setup, enter *Company Information* in the Quick Find box, then select **Company Information** and copy the ID located in the `Salesforce Organization ID`.
- b. For Customer Portals, from Setup, enter *Customer Portal Settings* in the Quick Find box, select **Customer Portal Settings**, click the name of the Customer Portal, and then copy the ID located in the `Portal ID`.
- c. For partner portals, from Setup, enter *Partners* in the Quick Find box, then select **Settings**. Next, click the name of the partner portal, and copy the ID located in the `Salesforce Portal ID`.

## EDITIONS

Available in: Salesforce Classic

Customer Portal is available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

Partner Portal is available in: **Enterprise, Performance,** and **Unlimited** Editions

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data



# DELEGATED AUTHENTICATION SINGLE SIGN-ON


You can integrate Salesforce with the authentication method of your choice using delegated authentication single sign-on (SSO). You can integrate with your LDAP (Lightweight Directory Access Protocol) server or authenticate with a token instead of a password. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Here's the process that Salesforce uses to authenticate users with delegated authentication SSO.

1. When a user tries to log in—either online or using the API—Salesforce validates the username and checks the user's permissions and access settings.
2. If the user has the "Is Single Sign-On Enabled" user permission, Salesforce doesn't validate the username and password. Instead, a web services call is made to the user's org asking it to validate the username and password.

 **Note:** Salesforce doesn't store, log, or view the password. It's disposed of immediately after the process completes.

3. The web services call passes the username, password, and sourceIp to your web service. Source Ip is the IP address where the login request originated. You must create and deploy an implementation of the web service that Salesforce servers can access.
4. Your web service implementation validates the passed information and returns either `true` or `false`.
5. If the response is `true`, the login process continues, a new session is generated, and the user proceeds to the app. If `false`, the user gets an error message that the username and password combination is invalid.

 **Note:** With delegated authentication, a user can experience a slight delay when logging in while the user account becomes available in the org.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

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

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data

## Configure Salesforce for Delegated Authentication

You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password. You must contact Salesforce to enable the delegated authentication feature before you can configure it in your org.

### 1. Build your SSO web service.

- a. In Salesforce, download the Web Services Description Language (WSDL) file `AuthenticationService.wsdl`. From Setup, enter `API` in the `Quick Find` box, then select **API**, then select **Download Delegated Authentication WSDL**.

The WSDL file describes the delegated authentication SSO service. Use the WSDL file to generate a server-side stub to which you add your SSO implementation. For example, in the WSDL2Java tool from Apache Axis, use the `--server-side` switch. With the .NET `wsdl.exe` tool, use the `/server` switch.

For a sample request and response, see [Sample SOAP Message for Delegated Authentication](#) on page 20.

- b. Add a link to your corporate intranet or other internal site that takes the authenticated user's credentials and passes them through an HTTP POST to the Salesforce login page.

Because Salesforce doesn't use the password field other than to pass it back to you, don't pass in a password. Instead, pass another authentication token, such as a Kerberos Ticket, so that your corporate passwords aren't passed to or from Salesforce.


You can configure the Salesforce delegated authentication authority to accept only a token or either a token or password. If the authority accepts only a token, Salesforce users can't log in to Salesforce directly because they can't create a valid token. However, many authorities support both tokens and passwords. In this case, users can log in to Salesforce through the login page.

When the Salesforce server passes the credentials back to you in the `Authenticate` message, verify them. Then the user can access the app.


### 2. In Salesforce, specify your org's SSO gateway URL. From Setup, enter `Single Sign-On` in the `Quick Find` box, select **Single Sign-On Settings**, and then click **Edit**. Enter the URL in the Delegated Gateway URL text box. For security reasons, Salesforce restricts outbound ports to one of the following.

- 80, which accepts only HTTP connections
- 443, which accepts only HTTPS connections
- 1024–66535, which accept HTTP or HTTPS connections

### 3. Optionally, select **Force Delegated Authentication Callout**.

-  **Note:** Select this option if you must record every login attempt. This option forces a callout to the SSO endpoint regardless of login restriction failures. If you don't select this option, a call isn't made to the SSO endpoint if the first login attempt fails due to login restrictions within the Salesforce org.

### 4. Enable the "Is Single Sign-On Enabled" permission.

-  **Important:** If single sign-on (SSO) is enabled for your org, API and desktop client users can log in to Salesforce unless their profile has IP address restrictions set and they try to log in from outside of the range defined. Also the SSO authority usually handles login lockout policies for users with the "Is Single Sign-On Enabled" permission. However, if the security token is enabled for your org,

#### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

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

#### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application
- AND
- Modify All Data

your org's login lockout settings determine how many times users can attempt to log in with an invalid security token before being locked out of Salesforce.

# SAMPLE DELEGATED AUTHENTICATION IMPLEMENTATIONS

Samples are available by downloading a ZIP file containing [sample code for .NET](#).

The samples are written in C# and authenticate users against Active Directory. The first sample is a simple implementation of delegated authentication. The second is a more complex sample that demonstrates a single-sign-on solution with an authentication token. Both samples use Microsoft .NET v1.1 and were deployed using IIS6 on a Windows 2003 server. Use the included `makefile` to build the samples.

## Sample 1

---

The example in `simple.asmx.cs` declares a new class, `SimpleAdAuth`, which is a web service with one method: `Authenticate`. Several attributes are declared on the method. The attributes control the formatting of the expected request and the generated response, and they set up the service to match the message definition in the WSDL. The implementation uses the passed credentials to try to connect to Active Directory via the LDAP provider. If it connects successfully, the credentials are good. Otherwise, the credentials are invalid.

## Sample 2

---

This more-complex example generates and verifies an authentication token rather than a password. The bulk of the implementation is in the `sso.asmx.cs` file, which defines a `SingleSignOn` class that generates an authentication token and implements the authentication service to verify that token. The generated token consists of a token number, expiration timestamp, and username. All the data is then encrypted and signed.

The verification process verifies the signature, decrypts the token, checks that it has not expired, and checks that the token number has not been previously used. (The token number and expiration timestamp are used to prevent replay attacks.) The file `gotosfdc.aspx` is an ASPX page designed to be deployed or linked to from an intranet site. This approach forces the user's authentication, generates a new authentication token for the user, and finally POSTs that token to the Salesforce login page along with a username that is mapped from the local NT username. The Salesforce login process sends the authentication token back to the service, which verifies the token and logs in the user. The file `intranet.aspx` is a simple page that links to `gotosfdc.aspx` so that you can see this process in action.

# SINGLE LOGOUT

With single logout (SLO), your users log out from one application, and are automatically logged out from other applications they are using.

For example, when Salesforce is the identity provider for connected applications, the user logs out from Salesforce and is automatically logged out of the other applications. Or, when a user is logged in to Salesforce from an identity provider using SAML, the user logs out of Salesforce and is automatically logged out of the identity provider, too. SLO can improve security and usability. Previously, your users had to remember to log out of each app separately.

To use SLO, the identity provider, service providers, and relying parties must be configured for single sign-on and registered for SLO.

Salesforce supports front-channel SLO, meaning your users are only logged out of their registered apps if they explicitly log out of one using their browsers. Having a session expire doesn't cause them to be logged out of the other apps registered for SLO.

Salesforce supports the following protocols:

- SAML SLO as an identity provider or service provider, initiated by either.
- OpenID Connect SLO as an identity provider or relying party, initiated by either.

Examples:

1. You want users to log in to Salesforce, then use connected apps to log in to other services. When they're ready to log out, they log out from Salesforce (or a configured service provider or relying party) and they're automatically logged out of all the configured connected apps and services. This behavior can be accomplished with the following:
  - SAML SLO for which Salesforce is the identity provider, and registered SAML connected apps are service providers
  - OpenID Connect SLO for which Salesforce is the identity provider, and registered OAuth connected apps are relying parties
2. You want users to log in to Salesforce using an external identity provider. The identity provider uses SAML or OpenID Connect to log the users in to a Salesforce org. When the users log out of the identity provider (or Salesforce) session, they're automatically logged out of both. This behavior can be accomplished with the following:
  - SAML SLO when Salesforce is the service provider connected to an external SAML identity provider
  - OpenID Connect SLO when Salesforce is the relying party connected to an external OpenID Connect provider

Implementing SLO brings several advantages to your org.

- Time savings—With SLO in place, users avoid manually logging out of connected apps. Fewer steps and no toggling through various apps saves time and reduces frustration.
- Increased security—Users don't have to remember to log out of any connected apps. When they log out of Salesforce, they are also logged out of the other apps. Even if a user leaves a desktop unattended, nobody can access these apps

SEE ALSO:

[Configure SAML Settings for Single Logout Where Salesforce Is the Service Provider](#)

[Configure SAML Settings for Single Logout Where Salesforce Is the Identity Provider](#)

[Configure OpenID Connect Settings for Single Logout Where Salesforce Is the Relying Party](#)

[Configure OpenID Connect Settings for Single Logout Where Salesforce Is the OpenID Connect Provider](#)

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration


To edit the settings:

- Customize Application  
AND  
Modify All Data

## Configure SAML Settings for Single Logout Where Salesforce Is the Service Provider

Configure SLO when Salesforce is the service provider connected to an external SAML identity provider. Users log in to an identity provider. The identity provider uses SAML to log the users in to a Salesforce org. When the users log out of the identity provider (or Salesforce) session, they're automatically logged out of both.

To use this feature:

- Enable My Domain.
  - Get the Issuer URL from the identity provider. This URL uniquely identifies your SAML identity provider. SAML assertions sent to Salesforce must match this value exactly in the `<saml:Issuer>` attribute of SAML assertions.
  - Get and save the certificate for validating signatures from the identity provider.
  - Get the single logout URL from the identity provider.
-  **Note:** Some identity providers don't support logout initiated by the service provider. In this case, do only step 6. Users will be able to log out of Salesforce when initiated by the identity provider. But, logging out of Salesforce won't necessarily log the user out of the identity provider session.
1. In Setup, enter *Single Sign-On Settings* in the `Quick Find` box, then select **Single Sign-On Settings**.
  2. In SAML Single Sign-On Settings, select **New**.
  3. On the **SAML Single Sign-On Settings** page, enter the required information and select **Single Logout Enabled**.
  4. For **Identity Provider Single Logout URL**, enter the SAML SLO endpoint of the identity provider. This URL is the endpoint where Salesforce sends LogoutRequests (when a logout is initiated by Salesforce), or LogoutResponses (when a logout is initiated by the identity provider). The identity provider gives you this endpoint.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data

The screenshot displays the 'SAML Single Sign-On Settings' configuration page in Salesforce. The page is titled 'SAML Single Sign-On Settings' and includes a 'Help for this Page' link. At the top, there are 'Save', 'Save & New', and 'Cancel' buttons. The configuration fields are as follows:

- Name: MyIdP
- API Name: MyIdP
- SAML Version: 2.0
- Issuer: https://myidp-developer
- Entity ID: https://mydomain-devel
- Identity Provider Certificate: Choose File (No file chosen)
- Request Signing Certificate: Generate self-signed certificate
- Request Signature Method: RSA-SHA256
- Assertion Decryption Certificate: Assertion not encrypted
- SAML Identity Type:
  - Assertion contains the User's Salesforce username
  - Assertion contains the Federation ID from the User object
  - Assertion contains the User ID from the User object
- SAML Identity Location:
  - Identity is in the NamelIdentifier element of the Subject statement
  - Identity is in an Attribute element
- Service Provider Initiated Request Binding:
  - HTTP POST
  - HTTP Redirect
- Identity Provider Login URL: (empty)
- Custom Logout URL: (empty)
- Custom Error URL: (empty)
- Single Logout Enabled:  (circled in red)
- Identity Provider Single Logout URL: https://myidp-developer-edition.my.salesforce.com/slo-logout
- Single Logout Request Binding:
  - HTTP POST
  - HTTP Redirect

At the bottom, there is a section for 'Just-in-time User Provisioning' with a 'User Provisioning Enabled' checkbox (unchecked) and an information icon. A legend indicates that a red exclamation mark icon represents 'Required Information'. The 'Save', 'Save & New', and 'Cancel' buttons are located at the bottom of the page.

5. Select the HTTP binding type to be used for service provider-initiated SLO. The binding type determines where to put the LogoutRequest or LogoutResponse in the SAML request. The value is base64 encoded.

**HTTP Redirect** — Sent in the querystring, deflated.

**HTTP POST** — Sent in the POST body, not deflated.

6. Provide your IdP with the Salesforce SP SLO endpoint. It is the **Logout URL** found under Your Organization in Endpoints on the **SAML Single Sign-On Settings** page. The format for the endpoint is `https://<domain>.my.salesforce.com/services/auth/sp/saml2/logout`, where `<domain>` is your org's My Domain name.

**SETUP**  
Single Sign-On Settings

**SAML Single Sign-On Settings** [Help for this Page](#)

[Back to Single Sign-On Settings](#)

[Edit](#) [Delete](#) [Clone](#) [Download Metadata](#) [SAML Assertion Validator](#)

Name		API Name	
SAML Version	2.0	Entity ID	
Issuer			
Identity Provider Certificate	CN= Certificate, OU=FOR DEMONSTRATION PURPOSES ONLY. DO NOT USE FOR PRODUCTION ENVIRONMENTS., O= L=San Francisco, ST=CA, C=US Expiration: 5 Nov 2041 04:30:27 GMT		
Request Signing Certificate	<a href="#">SelfSignedCert_13Oct2017_193802</a>		
Request Signature Method	RSA-SHA256		
Assertion Decryption Certificate	Assertion not encrypted		
SAML Identity Type	Username		
SAML Identity Location	Subject		
Service Provider Initiated Request Binding	HTTP Redirect		
Identity Provider Login URL			
Custom Logout URL			
Custom Error URL			
Single Logout Enabled	<input type="checkbox"/>		

**Just-in-time User Provisioning**

User Provisioning Enabled

**Endpoints**  
View SAML endpoints for your organization, communities, or custom domains.

Your Organization

Login URL	https://.my.salesforce.com?so=00DB0000000JCNM
Logout URL	https://.my.salesforce.com/services/auth/sp/saml2/logout
OAuth 2.0 Token Endpoint	https://.my.salesforce.com/services/oauth2/token?so=00DB0000000JCNM

[Edit](#) [Delete](#) [Clone](#) [Download Metadata](#) [SAML Assertion Validator](#)

If the org is a Salesforce Community, the Logout URL for the community appears on the same page.

If you participate in the Logout Event Triggers pilot program, you can configure an Apex trigger that responds to logout events.

SEE ALSO:

[Single Logout](#)

[Salesforce Help: Create Logout Event Triggers \(Beta\)](#)




## Configure SAML Settings for Single Logout Where Salesforce Is the Identity Provider

Configure SLO when Salesforce is the identity provider connected to an external SAML service provider. Users log in to Salesforce. Salesforce uses SAML to log in users to the service provider through a connected app. When the users log out of the service provider (or Salesforce) session, they're automatically logged out of both.


To use this feature:

- Enable My Domain.
- Make sure that the service provider supports SAML SLO.
- Get the SAML SLO endpoint from the service provider.
- Find out the HTTP binding type from the service provider.

 **Note:** Some service providers don't support initiating single logout. In this case, skip step 6. Users are logged out of the service provider when initiated by Salesforce. But, logging out of the service provider won't necessarily log the user out of Salesforce.

This implementation uses connected apps. You can configure SLO when you create and edit a connected app as a developer, and distribute it to other orgs. Or, you can create and manage SLO for a connected app within your org as an administrator. Changes to the SLO configuration in the connected app management page are not propagated back to the page when you're editing a connected app as a developer. As you change settings through connected app management pages, manually copy settings to the app creation page, if desired.

1. For an existing connected app: In Setup, enter `apps` in the `Quick Find` box, then select **Manage Connected Apps**.
2. Next to the connected app that you want to configure for SLO, click **Edit**. You are now editing the connected app configuration, even though the path here was through **Manage Connected Apps**.
3. Under SAML Service Provider Settings, select **Enable Single Logout**.



4. For Single Logout URL, enter the SAML SLO endpoint of the connected app service provider (SP). The URL must start with `https://`. This URL is the endpoint where Salesforce sends LogoutRequests (when a logout is initiated by Salesforce), or LogoutResponses (when a logout is initiated by the service provider). The service provider gives you this endpoint.
5. Select the HTTP binding type for SLO. The binding type determines where to put the LogoutRequest or LogoutResponse in the SAML request. The value is base64 encoded. The service provider gives you this information.

**HTTP Redirect** — Sent in the querystring, deflated.

**HTTP POST** — Sent in the POST body, not deflated.

6. Provide your service provider with the Salesforce identity provider SLO endpoint. With this endpoint, the service provider can initiate SLO. It's listed in the **Single Logout Endpoint** under SAML Login Information on the Connected App Detail page, and in the SAML

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

### USER PERMISSIONS

To view the settings:

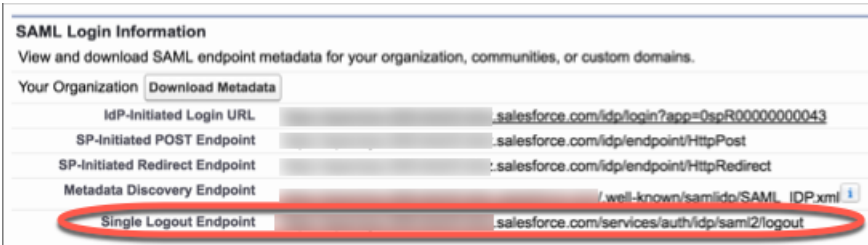
- View Setup and Configuration

To edit the settings:

- Customize Application AND Modify All Data

Metadata Discovery Endpoint. The format for the endpoint is

`https://<domain>.my.salesforce.com/services/auth/idp/saml2/logout`, where `<domain>` is your org's My Domain name.



SAML Login Information	
View and download SAML endpoint metadata for your organization, communities, or custom domains.	
Your Organization	<a href="#">Download Metadata</a>
IdP-Initiated Login URL	...salesforce.com/idp/login?app=0spR00000000043
SP-Initiated POST Endpoint	...salesforce.com/idp/endpoint/HttpPost
SP-Initiated Redirect Endpoint	...salesforce.com/idp/endpoint/HttpRedirect
Metadata Discovery Endpoint	.../well-known/samlidp/SAML_IDP.xml ⓘ
Single Logout Endpoint	...salesforce.com/services/auth/idp/saml2/logout

If you participate in the Logout Event Triggers pilot program, you can configure an Apex trigger that responds to logout events.

SEE ALSO:

[Single Logout](#)


[Salesforce Help: Create Logout Event Triggers \(Beta\)](#)

# Configure OpenID Connect Settings for Single Logout Where Salesforce Is the Relying Party

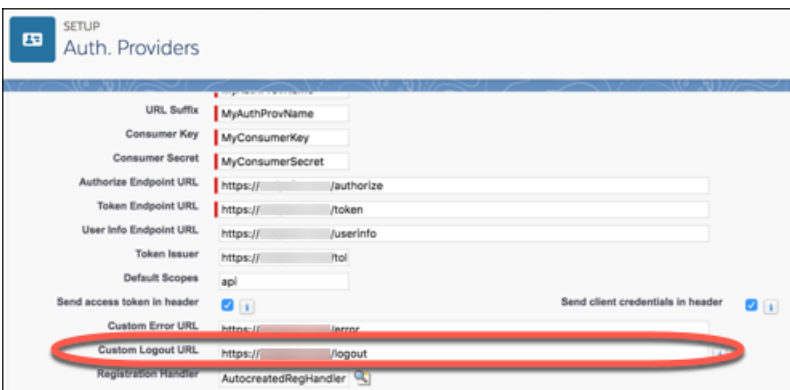
Configure SLO when authentication providers use OpenID Connect to give users access to Salesforce as the relying party. Users log in to Salesforce through the authentication provider. When the users log out of Salesforce (or the authentication provider) session, they're automatically logged out of both.

To use this feature:

- Enable My Domain.
- Make sure that the authentication provider supports OpenID Connect SLO.
- Set up the authentication provider.
- Get the OpenID Connect SLO logout endpoint from the authentication provider.

 **Note:** Some authentication providers don't support logout initiated by the relying party. In this case, do only step 5. Users will be able to log out of Salesforce when initiated by the authentication provider. But, logging out of Salesforce won't necessarily log the user out of the authentication provider session.

1. In Setup, enter *Auth. Providers* in the **Quick Find** box, then select **Auth. Providers**.
2. Next to the auth provider that you want to configure for SLO, click **Edit**.
3. Under **Auth. Provider Edit**, enter the logout endpoint from the authentication provider in **Custom Logout URL**. With this endpoint, Salesforce can initiate SLO. The Custom Logout URL must be an absolute URL and start with *http://* or *https://*.



4. Click **Save**.
5. Provide your authentication provider with the Salesforce SLO endpoint. With this endpoint, the authentication provider can initiate SLO. It's the **Single Logout URL** found under Salesforce Configuration on the Auth. Provider detail page. The format for the endpoint is *https://<domain>.my.salesforce.com/services/auth/rp/oidc/logout*, where *<domain>* is your org's My Domain name.

## EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All** Editions

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

## USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application  
AND  
Modify All Data

The screenshot shows the Salesforce Setup page for 'Auth. Providers'. The page is titled 'Auth. Provider' and includes a 'Help for this Page' link. Below the title, there is a 'Back to List: Custom Apps' link. The main section is 'Auth. Provider Detail', which contains a table of configuration fields. The fields are: Auth. Provider ID, Provider Type, Name, URL Suffix, Consumer Key, Consumer Secret (with a 'Click to reveal' link), Authorize Endpoint URL (https://login.salesforce.com/services/oauth2/authorize), Token Endpoint URL (https://login.salesforce.com/services/oauth2/token), Default Scopes (api), Include identity organization's organization ID for third-party account linkage (checkbox), Custom Error URL (/error), Custom Logout URL (/logout), Registration Handler, Execute Registration As (Portal), and Icon URL (https://login.salesforce.com/icons/google-grey.png). Below this is the 'Salesforce Configuration' section, which includes: Test-Only Initialization URL, Existing User Linking URL, OAuth-Only Initialization URL, Callback URL, and Single Logout URL (https://...my.salesforce.com/services/auth/rp/oidc/logout). The 'Single Logout URL' field is circled in red. At the bottom of the page, there are 'Edit', 'Delete', and 'Clone' buttons.

Auth. Provider Detail	
Auth. Provider ID	
Provider Type	
Name	
URL Suffix	
Consumer Key	
Consumer Secret	<a href="#">Click to reveal</a>
Authorize Endpoint URL	https://login.salesforce.com/services/oauth2/authorize
Token Endpoint URL	https://login.salesforce.com/services/oauth2/token
Default Scopes	api
Include identity organization's organization ID for third-party account linkage	<input type="checkbox"/>
Custom Error URL	/error
Custom Logout URL	/logout
Registration Handler	
Execute Registration As	Portal
Icon URL	https://login.salesforce.com/icons/google-grey.png

Salesforce Configuration	
Test-Only Initialization URL	https://login.salesforce.com/services/auth/test/
Existing User Linking URL	https://login.salesforce.com/services/auth/link/
OAuth-Only Initialization URL	https://login.salesforce.com/services/auth/oauth/
Callback URL	https://login.salesforce.com/services/auth/callback/
Single Logout URL	https://...my.salesforce.com/services/auth/rp/oidc/logout

If you participate in the Logout Event Triggers pilot program, you can configure an Apex trigger that responds to logout events.

SEE ALSO:

[Single Logout](#)

[Salesforce Help: External Authentication Providers](#)

[Salesforce Help: Create Logout Event Triggers \(Beta\)](#)

## Configure OpenID Connect Settings for Single Logout Where Salesforce Is the OpenID Connect Provider

Configure SLO when Salesforce provides authentication for users to access a relying provider using OpenID Connect. Users log in to Salesforce. Salesforce uses OpenID Connect to authenticate users for the relying party through a connected app. When the users log out of the relying party (or Salesforce) session, they're automatically logged out of both.

To use this feature:

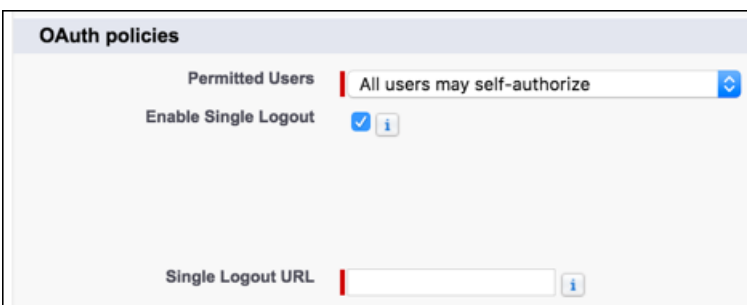
- Enable My Domain.
- Make sure the relying party supports OpenID Connect SLO.
- Get the OpenID Connect SLO logout endpoint from the relying party.

This implementation uses connected apps. You can configure SLO when you create and edit a connected app as a developer, and distribute it to other orgs. Or, you can create and manage SLO for a connected app within your org as an administrator. Changes to the SLO configuration in the connected app management page are not propagated back to the page when you're editing a connected app as a developer. As you change settings through connected app management pages, manually copy settings to the app creation page, if desired.

Also, after the initial creation of the connected app, changes to the SLO configuration for the connected app development page do not propagate to the administration page, automatically.

These steps edit an existing connected app. The fields are the same when you create, or manage, a connected app.

1. In Setup, enter *apps* in the *Quick Find* box, then select **Manage Connected Apps**.
2. Next to the connected app that you want to configure for SLO, click **Edit**.
3. Under **OAuth Policies**, select **Enable Single Logout**.



4. For **Single Logout URL**, enter the OpenID Connect SLO endpoint of the connected app's relying party. This endpoint is where Salesforce sends a logout request when users log out of Salesforce. The relying party provides you with this endpoint. The Single Logout URL must be an absolute URL and start with *https://*.
5. Use the [OpenID Connect Discovery Endpoint](#) to provide your relying party with the Salesforce identity provider SLO endpoint. With this endpoint, the relying party can initiate SLO. It's found in *https://<domain>.my.salesforce.com/.well-known/openid-configuration*, where *<domain>* is your org's My Domain name. The format for the endpoint is *https://<domain>.my.salesforce.com/services/auth/idp/oidc/logout*, also where *<domain>* is your org's My Domain name.

### EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

### USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application AND Modify All Data

```
{  
  "end_session_endpoint": "https://[redacted].my.salesforce.com/services/auth/idp/oidc/logout",  
  "frontchannel_logout_supported": true,  
  "frontchannel_logout_session_supported": false,  
  "issuer": "https://[redacted].my.salesforce.com",  
  "authorization_endpoint": "https://[redacted].my.salesforce.com/services/oauth2/authorize",  
  "token_endpoint": "https://[redacted].my.salesforce.com/services/oauth2/token",  
  "revocation_endpoint": "https://[redacted].my.salesforce.com/services/oauth2/revoke",  
  "userinfo_endpoint": "https://[redacted].my.salesforce.com/services/oauth2/userinfo",  
  "jwks_uri": "https://[redacted].my.salesforce.com/id/keys",  
  "scopes_supported": [  

```

If you participate in the Logout Event Triggers pilot program, you can configure an Apex trigger that responds to logout events.

SEE ALSO:

[Single Logout](#)

[Salesforce Help: Create Logout Event Triggers \(Beta\)](#)

# FREQUENTLY ASKED QUESTIONS

## How do I enable single sign-on?

Salesforce offers the following ways to use SSO.

- Federated authentication using Security Assertion Markup Language (SAML) lets you send authentication and authorization data between affiliated but unrelated web services. You can log in to Salesforce from a client app. Salesforce enables federated authentication for your org automatically.
- Delegated authentication SSO integrates Salesforce with an authentication method that you choose. You can integrate authentication with your LDAP (Lightweight Directory Access Protocol) server or use a token instead of a password for authentication. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Delegated authentication offers the following benefits.

- Uses a stronger form of user authentication, such as integration with a secure identity provider
- Makes your login page private and accessible only behind a corporate firewall
- Differentiates your org from all other companies that use Salesforce to reduce phishing attacks

You must contact Salesforce to enable delegated authentication before you can configure it on your org.

- Authentication providers let your users log in to your Salesforce org using their login credentials from an external service provider. Salesforce supports the OpenID Connect protocol, which lets users log in from any OpenID Connect provider, such as Google, PayPal, and LinkedIn. When an authentication provider is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

## Where in Salesforce do I configure single sign-on?

For delegated authentication single sign-on:

- To access the WSDL, from Setup, enter *API* in the **Quick Find** box, select **API**, and then select **Download Delegated Authentication WSDL**.
- To specify your organization's single sign-on gateway URL, from Setup, enter *Single Sign-On Settings* in the **Quick Find** box, select **Single Sign-On Settings**, and then select **Edit**.
- To enable the "Is Single Sign-On Enabled" user permission for your single sign-on users, from Setup, enter *Permission Sets* in the **Quick Find** box, then select **Permission Sets**.

For federated authentication using SAML:

- From Setup, enter *Single Sign-On Settings* in the **Quick Find** box, select **Single Sign-On Settings**, and then click **Edit**.

## How are passwords reset when single sign-on has been implemented?

Password reset is disabled for single sign-on users who use delegated authentication because Salesforce no longer manages their passwords. Users who try to reset their passwords in Salesforce will be directed to their Salesforce administrator.

## Where can I view single sign-on login errors?

For delegated authentication, administrators with the "Modify All Data" permission can view the twenty-one most recent single sign-on login errors for your organization from Setup by entering *Delegated Authentication Error History* in the **Quick Find** box, then selecting **Delegated Authentication Error History**. For each failed login, you can view the user's username, login time, and the error. For federated authentication, administrators can view login errors from Setup by entering *Login History* in the **Quick Find** box, then selecting **Login History**.

### Where can I find entries about login history for a failed SAML login attempt?

When Salesforce cannot find the user in your assertion or cannot associate the provided user ID with a user in Salesforce, an entry is inserted in the login history. To see the login history, from Setup, enter *Login History* in the **Quick Find** box, then select **Login History**.

### Does single sign-on work outside my corporate firewall?

Yes, single sign-on can work outside your corporate firewall. When users are outside the corporate firewall, they can use their network passwords to log in to Salesforce. Alternately, you can require that users must first be connected to your corporate network in order to log in.

### Can I validate the SAML response sent by my identity provider?

Yes. After you have configured single sign-on, you can access the SAML Validation page from Setup, by clicking **SAML Validation** on the Single Sign-On Settings page. If a user tries to log in to Salesforce and fails, the invalid SAML assertion is used to automatically populate the SAML Assertion Validator if possible. On the SAML Validation page, if the SAML assertion is not automatically populated, you can enter either an XML- or base64-encoded SAML response that you've received from your service provider. Salesforce validates the response against the values provided during single sign-on setup, and provides detailed information about the response.

### Can I configure a start page and logout page that are specific to my company?

Yes.

You can customize the start, error, login, and logout pages for single sign-on users using SAML 1.1 or 2.0. As part of your configuration, decide the following:

- If your identity provider uses SAML 1.1, the URL to direct the user to when single sign-on successfully completes (known as the start page). This URL can be absolute, such as `https://yourInstance.salesforce.com/001/o` or it can be relative, such as `/001/o`. This URL must be an endpoint that accepts SAML authentication requests.

In SAML 2.0, the start page is the page the user attempted to access before they were authenticated. The SAML 2.0 start page must support Sp-init single sign-on.

If you are using SAML 2.0, you can also use the `RelayState` parameter to control where users get redirected after a successful login.

- The single sign-on start page where Salesforce sends a SAML request to start the login sequence.

We recommend that if you specify a single sign-on start page that you also specify a logout page. When you specify a logout page, when a user clicks logout or if a user's session expires, the user is redirected to that page. If you don't specify a logout page, the user is redirected to the general Salesforce login page.

- The URL to direct the user to when they click the Logout link in Salesforce (known as the logout page). The default is `https://login.salesforce.com`, unless MyDomain is enabled. If My Domain is enabled, the default is `https://customdomain.my.salesforce.com`.

See [Customize SAML Start, Error, Login, and Logout Pages](#) on page 12.

### Does Salesforce delegated authentication support SAML tokens?

Yes, SAML tokens can be used with the [sample delegated authentication implementations](#) using the listener validating the token.

### Can delegated authentication single sign-on work with Connect Offline?

Yes, delegated authentication can work with Connect Offline if it is set up to work with both tokens and passwords. In this case, users should use their network password to access Connect Offline.



# INDEX

## C

Canvas Apps [52](#)

## D

Delegated authentication  
    configuring single sign-on [62](#)  
    sample implementations [64](#)  
    single sign-on [61](#)

## E

Error page  
    customizing in SAML [12](#)

## I

Identity provider  
    values [8](#)  
Identity providers  
    examples [41, 47](#)

## J

Just-in-time provisioning  
    example SAML assertions [13](#)  
Just-in-Time provisioning  
    community requirements [34](#)  
    portal requirements [31](#)  
    requirements [29](#)  
Just-in-Time provisioning errors [38](#)

## L

Logging in  
    SAML start page [12](#)  
Logging out  
    SAML [12](#)

## P

Portals  
    single sign-on [60](#)

## S

Salesforce as Identity Provider:  
    Canvas Apps [52](#)  
SAML  
    about [2](#)  
    custom error page [12](#)

### SAML (*continued*)

    example assertions [13](#)  
    Just-in-Time for communities [34](#)  
    Just-in-Time for portals [31](#)  
    Just-in-Time provisioning [29](#)  
    Just-in-Time provisioning errors [38](#)  
    Just-in-Time provisioning requirements [29](#)  
    login history [28](#)  
    login page [12](#)  
    logout page [12](#)  
    prerequisites [7](#)  
    single logout [66, 69, 71, 73](#)  
    single sign-on [3](#)  
    start page [12](#)  
    validating single sign-on [25](#)  
    validation errors [26](#)  
    viewing single sign-on [24](#)

### Security

    Just-in-Time for communities [34](#)  
    Just-in-Time for portals [31](#)  
    Just-in-Time provisioning [29](#)  
    Just-in-Time provisioning requirements [29](#)  
    portals single sign-on [60](#)

### Service providers

    examples [41, 47](#)

### Single logout

    overview [65](#)  
    SAML [66, 69, 71, 73](#)

### Single sign-on

    best practices [44](#)  
    configuring delegated authentication [62](#)  
    debugging [25](#)  
    delegated authentication [61, 64](#)  
    example SAML assertions [13](#)  
    FAQ [75](#)  
    identity provider values [8](#)  
    login history [28](#)  
    overview [1](#)  
    portals [60](#)  
    prerequisites [7](#)  
    SAML [3](#)  
    SAML validation [25](#)  
    viewing [24](#)

Single sign-on to Canvas [52](#)