Single Sign-On Implementation Guide

Salesforce, Spring '16



@salesforcedocs

Last updated: March 3, 2016

© Copyright 2000–2016 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

Single Sign-On 1
About SAML 3
Configuring SAML Settings for Single Sign-On 4
Working With Your Identity Provider 8
Identity Provider Values 9
Customize SAML Start, Error, Login, and Logout Pages 13
Example SAML Assertions 14
Viewing Single Sign-On Settings 25
Validating SAML Settings for Single Sign-On 26
SAML Assertion Validation Errors 27
Reviewing the SAML Login History 29
About Just-in-Time Provisioning for SAML 30
Just-in-Time Provisioning Requirements 30
Just-in-Time Provisioning for Portals 32
Just-in-Time Provisioning for Communities 35
Just-in-Time Provisioning Errors 38
Best Practices for Implementing Single Sign-On 41
Enable Single Sign-On for Portals 44
Understanding Delegated Authentication Single Sign-On 45
Configuring Salesforce for Delegated Authentication 46
Sample Delegated Authentication Implementations 48
Frequently Asked Questions 49
Index 51

SINGLE SIGN-ON

Single sign-on allows users to access all authorized network resources without having to log in separately to each resource. You validate usernames and passwords against your corporate user database or other client application rather than having separate user passwords managed by Salesforce.

Salesforce offers the following ways to use single sign-on:

- Federated authentication using Security Assertion Markup Language (SAML) allows you to send authentication and authorization data between affiliated but unrelated Web services. This enables you to sign on to Salesforce from a client application. Federated authentication using SAML is enabled by default for your organization.
- Delegated authentication single sign-on enables you to integrate Salesforce with an authentication method that you choose. This enables you to integrate authentication with your LDAP (Lightweight Directory Access Protocol) server, or perform single sign-on by authenticating using a token instead of a password. You manage delegated authentication at the permission level, allowing some users to use delegated authentication, while other users continue to use their Salesforce-managed password. Delegated authentication is set by permissions, not by organization.

  The primary reasons for using delegated authentication include:

  - Using a stronger type of user authentication, such as integration with a secure identity provider
  - Making your login page private and accessible only behind a corporate firewall
  - Differentiating your organization from all other companies that use Salesforce in order to reduce phishing attacks

  You must request that this feature be enabled by Salesforce. Contact Salesforce to enable delegated authentication single sign-on for your organization.
- Authentication providers let your users log in to your Salesforce organization using their login credentials from an external service provider. Salesforce supports the OpenId Connect protocol that allows users to log in from any OpenID provider such as Google, PayPal, LinkedIn and other services supporting OpenID Connect. When authentication providers are enabled, Salesforce does not validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

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"

When you have an external identity provider, and configure single sign-on for your Salesforce organization, Salesforce is then acting as a *service provider*. You can also enable Salesforce as an identity provider, and use single sign-on to connect to a different service provider. Only the service provider needs to configure single sign-on.

The Single Sign-On Settings page displays which version of single sign-on is available for your organization. To learn more about the single sign-on settings, see Configuring SAML Settings for Single Sign-On. For more information about SAML and Salesforce security, see the *Security Implementation Guide*.

Benefits of Single Sign-On

Implementing single sign-on can offer the following advantages to your organization:

- **Reduced Administrative Costs:** With single sign-on, users only need to memorize a single password to access both network resources or external applications and Salesforce. When accessing Salesforce from inside the corporate network, users are logged in seamlessly, without being prompted to enter a username 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 administrators receive fewer requests to reset forgotten passwords.
- **Leverage Existing Investment:** Many companies use a central LDAP database to manage user identities. By delegating Salesforce authentication to this system, when a user is removed from the LDAP system, they can no longer access Salesforce. Consequently, users who leave the company automatically lose access to company data after their departure.
- **Time Savings:** On average, a user takes five to 20 seconds to log in to an online application; longer if they mistype their username or password and are prompted to reenter them. With single sign-on in place, the need to manually log in to Salesforce is avoided. These saved seconds 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 on a regular basis. For example, users can send email messages that contain links to information in Salesforce such as records and reports. When the recipients of the email message click the links, the corresponding Salesforce page opens automatically.
- **Increased Security:** Any password policies that you have established for your corporate network will also be 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.

About SAML

Security Assertion Markup Language (SAML) is an XML-based standard that allows you to communicate authentication decisions between one service and another. It underlies many Web single sign-on solutions. Salesforce supports SAML for single sign-on into Salesforce from a corporate portal or identity provider.

Much of the work to set up single sign-on using SAML occurs with your identity provider:

1. Establish a SAML identity provider and gather information about how they will connect to Salesforce. This is the provider that will send single sign-on 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 Configuring SAML Settings for Single Sign-On. This is the only step that takes place in Salesforce.

Your identity provider should send 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 allows single sign-on if the assertion is true.

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 may need to obtain a SAML assertion from your identity provider.

If your users are having problems using SAML to login, 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 your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Salesforce organization or community.

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"

CONFIGURING SAML SETTINGS FOR SINGLE SIGN-ON

From this page, you can configure your organization 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.

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"

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 new configuration, as follows.
   - **New** - Specify all settings manually.
   - **New from Metadata File** - Import SAML 2.0 settings from a 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 located 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 organization.

   Salesforce inserts the corresponding **API Name** value, which you can customize if necessary.
7. Enter the `Issuer`. This is often referred to as the entity ID for the identity provider.

8. If your Salesforce organization 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.
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 organization 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 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 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.
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**.

If your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Salesforce organization or community.

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 (this applies to organizations created before the Summer '13 release that are not using SAML 1.1).To

enable multiple 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 organization'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 SamlJitHandler 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

- 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 may be unable 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 may have to obtain a SAML assertion from your identity provider first.

If your users are having problems using SAML to login, 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 this 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: • SAML 1.1 • SAML 2.0
`Issuer`	The Entity ID—a URL that uniquely identifies your SAML identity provider. SAML assertions sent to Salesforce must match this value exactly in the `<saml:Issuer>` attribute of SAML assertions.
`Entity ID`	The issuer in SAML requests generated by Salesforce, and is also the expected audience of any inbound SAML Responses. If you don't have domains deployed, this value is always `https://saml.salesforce.com`. 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 *`Single Sign-On Settings`* in the `Quick Find` box, then select **Single Sign-On Settings**.
`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 `RSA-SHA1` or `RSA-SHA256`.
`SAML Identity Type`	The element in a SAML assertion that contains the string that identifies a Salesforce user. Values are: **`Assertion contains User's Salesforce username`** 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
	Assertion contains the Federation ID from the User object 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. **Assertion contains the User ID from the User object** 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.
SAML Identity Location	The location in the assertion where a user should be identified. Values are: **Identity is in the NameIdentifier element of the Subject statement** The Salesforce Username or FederationIdentifier is located in the <Subject> statement of the assertion. **Identity is in an Attribute element** The Salesforce Username or FederationIdentifier is specified in an <AttributeValue>, located in the <Attribute> of the assertion.
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 Assertions and Protocols SAML 2.0 specification.
Service Provider Initiated Request Binding	If you're using My Domain, chose the binding mechanism your identity provider requests for your SAML messages. Values are: **HTTP POST** HTTP POST binding sends SAML messages using base64-encoded HTML forms. **HTTP Redirect** HTTP Redirect binding sends base64-encoded and URL-encoded SAML messages within URL parameters. No matter what request binding is selected, the SAML Response will always use HTTP POST binding.
Identity Provider Login URL	For SAML 2.0 only: The URL where Salesforce sends a SAML request to start the login sequence. If you have domains 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: http://mydomain.my.salesforce.com?login. Note: 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.

Field	Description
`Identity Provider Logout URL`	For SAML 2.0 only: The URL to direct the user to when they click the **Logout** link in Salesforce. The default is `http://www.salesforce.com`. Note: 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://na1.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://na1.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 theSalesforce 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>
        AzID5hhJeJlG2llUDvZswNUrlrPtR7S37QYH2W+Un1n8c6kTC
        Xr/lihEKPcA2PZt86eBntFBVDWTRlh/W3yUgGOqQBJMFOVbhK
        M/CbLHbBUVT5TcxIqvsNvIFdjIGNkf1W0SBqRKZOJ6tzxCcLo
        9dXqAyAUkqDpX5+AyltwrdCPNmncUM4dtRPjI05CL1rRaGeyX
        3kkqOL8p0vjm0fazU5tCAJLbYuYgU1LivPSahWNcpvRSlCI4e
        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/CbLHbBUVT5TcxIqvsNvIFdjIGNkf1W0SBqRKZOJ6tzxCcLo
            9dXqAyAUkqDpX5+AyltwrdCPNmncUM4dtRPjI05CL1rRaGeyX
            3kkqOL8p0vjm0fazU5tCAJLbYuYgU1LivPSahWNcpvRSlCI4e
            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">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">00DD0000000F7Z5
      </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://ap1.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>
```

```
         <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">000000000000000
   </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>
```

VIEWING SINGLE SIGN-ON SETTINGS

After you have configured your Salesforce organization to use SAML, you can view the single sign-on settings. From Setup, enter *Single Sign-On Settings* in the Quick Find box, then select **Single Sign-On Settings**.

This page lists the details of your SAML configuration. Most of these fields are the same as the fields on the page where you configured SAML. The following fields contain information automatically generated by completing the configuration. The available fields depend on your configuration.

Field	Description
Salesforce Login URL	For SAML 2.0 only. If you select "Assertion contains User's Salesforce username" for SAML User ID Type and "User ID is in the NameIdentifier element of the Subject statement" for SAML User ID Location, this URL is the URL associated with login for the Web single sign-on OAuth assertion flow.
Salesforce Logout URL	For SAML 2.0. Displays the Salesforce logout URL that the user is directed to after he or she logs off. This URL is only used if no value is specified for Identity Provider Logout URL.
OAuth 2.0 Token Endpoint	For SAML 2.0 only: The ACS URL used with enabling Salesforce as an identity provider in the Web single sign-on 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 organization using a SAML assertion provided by your identity provider.
- If your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Salesforce organization or community.

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: 00Dx0000000BQlI
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

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

- `Provision Version` 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 standard fields are supported.

Fields	Required	Comments
`AboutMe`		
`Alias`		If not present, a default is derived from FirstName and LastName.
`CallCenter`		
`City`		
`CommunityNickname`		If not present, a default is derived from the UserName.
`CompanyName`		
`Country`		
`DefaultCurrencyIsoCode`		Derived from organization settings.
`DelegatedApproverId`		
`Department`		
`Division`		
`Email`	Y	For example, `User.Email=test2@salesforce.com`
`EmailEncodingKey`		If not present, a default is derived from the organization settings.
`EmployeeNumber`		
`Extension`		
`Fax`		
`FederationIdentifier` (insert only)		If present, it must match the SAML subject, or the SAML subject is taken instead. Can't be updated with SAML.
`FirstName`		
`ForecastEnabled`		
`IsActive`		
`LastName`	Y	
`LanguageLocaleKey`		
`LocaleSidKey`		If not present, a default is derived from the organization settings.
`Manager`		
`MobilePhone`		
`Phone`		

Fields	Required	Comments
`ProfileId`	Y	For example, `User.ProfileId=Standard User`
`ReceivesAdminInfoEmails`		
`ReceivesInfoEmails`		
`State`		
`Street`		
`TimeZoneSidKey`		If not present, a default is derived from the organization settings.
`Title`		
`Username` (insert only)	Y	For example, `User.Username=test2@test.com`. Can't update using SAML.
`UserRoleId`		Defaults to "no role" if blank.
`Zip`		

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 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.

  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 fields supported for regular SAML JIT users, these fields are supported for accounts.

Fields	Required	Comments
`Billing`		Street|City|State|PostalCode|Country
`AnnualRevenue`		
`Description`		
`Fax`		

Fields	Required	Comments
`FederationIdentifier` (insert only)	Y	If present, it must match the SAML subject or the SAML subject is taken instead. Can't be updated using SAML.
`IsCustomerPortal`		
`IsPartner`		
`NumberOfEmployees`		
`Ownership`		
`Phone`		
`Portal Role`	Y	Use `Worker` for all portal users.
`Rating`		
`Street`		
`TickerSymbol`		
`UserRoleId`		Defaults to "no role" if blank.
`Website`		
`Zip`		

In addition to the standard fields supported for regular SAML JIT users, these fields are supported for contacts.

Fields	Required	Comments
`Birthdate`		
`CanAllowPortalSelfReg`		Name|Phone
`Department`		
`Description`		
`DoNotCall`		
`Fax`		
`HasOptedOutofEmail`		
`HasOptedOutofFax`		
`HomePhone`		
`LeadSource`		
`Mailing`		Street|City|State|PostalCode|Country
`MobilePhone`		
`Owner`		

Fields	Required	Comments
`Other`		Street|City|State|PostalCode|Country
`OtherPhone`		
`Phone`		
`Salutation`		
`Title`		

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 Configuring 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=00DD0000000JsCM"` where `acme.force.com/customers` is the community home page and `00DD0000000JsCM` 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=00DD0000000JsCM&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 `FederationIdentifier` field of an existing user record.
  1. If a matching user record is found, Salesforce uses the attributes in the SAML assertion to update the specified fields.
  2. If a user with a matching user record isn't found, then Salesforce searches the contacts under the specified `Account ID` (`Contact.Account` or `Account.AccountNumber`) for a match based on the Contact ID (`User.ContactId`) or email (`Contact.Email`).
     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 both the `Account.AccountNumber` and `Account.Name` specified in the SAML assertion, or updates the existing contact and user records for the account if a match is found..
        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 a new account record, inserts a new contact record, and inserts a new user record 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 `Federation ID`, 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.

Salesforce attempts to match the `Federation ID` in the subject of the SAML assertion to the `FederationIdentifier` field of an existing user record.

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 fields supported for regular SAML JIT users, these fields are supported for accounts.

Fields	Required	Comments
`Billing`		Street|City|State|PostalCode|Country
`AnnualRevenue`		
`Description`		
`Fax`		
`FederationIdentifier` (insert only)	Y	If present, it must match the SAML subject or the SAML subject is taken instead. Can't be updated using SAML.
`IsCustomerPortal`		
`IsPartner`		
`NumberOfEmployees`		
`Ownership`		
`Phone`		
`Portal Role`		
`Rating`		
`Street`		
`TickerSymbol`		
`UserRoleId`		Defaults to "no role" if blank.
`Website`		
`Zip`		

In addition to the standard fields supported for regular SAML JIT users, these fields are supported for contacts.

Fields	Required	Comments
`Birthdate`		
`CanAllowPortalSelfReg`		Name|Phone
`Department`		
`Description`		
`DoNotCall`		
`Fax`		
`HasOptedOutofEmail`		
`HasOptedOutofFax`		

Fields	Required	Comments
HomePhone		
LeadSource		
Mailing		Street|City|State|PostalCode|Country
MobilePhone		
Owner		
Other		Street|City|State|PostalCode|Country
OtherPhone		
Phone		
Salutation		
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

Code	Description	Error Details
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
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

Code	Description	Error Details
42	Updating a contact with Person Account isn't supported	UNSUPPORTED_CONTACT_PERSONACCT_UPDATE

BEST PRACTICES FOR IMPLEMENTING SINGLE SIGN-ON

Salesforce offers the following ways to use single sign-on:

- Federated authentication using Security Assertion Markup Language (SAML) allows you to send authentication and authorization data between affiliated but unrelated Web services. This enables you to sign on to Salesforce from a client application. Federated authentication using SAML is enabled by default for your organization.
- Delegated authentication single sign-on enables you to integrate Salesforce with an authentication method that you choose. This enables you to integrate authentication with your LDAP (Lightweight Directory Access Protocol) server, or perform single sign-on by authenticating using a token instead of a password. You manage delegated authentication at the permission level, allowing some users to use delegated authentication, while other users continue to use their Salesforce-managed password. Delegated authentication is set by permissions, not by organization.

  The primary reasons for using delegated authentication include:

  - Using a stronger type of user authentication, such as integration with a secure identity provider
  - Making your login page private and accessible only behind a corporate firewall
  - Differentiating your organization from all other companies that use Salesforce in order to reduce phishing attacks

  You must request that this feature be enabled by Salesforce. Contact Salesforce to enable delegated authentication single sign-on for your organization.
- Authentication providers let your users log in to your Salesforce organization using their login credentials from an external service provider. Salesforce supports the OpenId Connect protocol that allows users to log in from any OpenID provider such as Google, PayPal, LinkedIn and other services supporting OpenID Connect. When authentication providers are enabled, Salesforce does not 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.

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"

Delegated Authentication Best Practices

Consider the following best practices when implementing delegated authentication single sign-on for your organization.

- Your organization's implementation of the Web service must be accessible by Salesforce servers. This means 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 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 cannot connect, or the request takes longer than 10 seconds to process, the login attempt fails. An error is reported to the user indicating that his or her 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 to ensure accuracy.

- For security reasons, you should make your Web service available by TLS. You must use a certificate from a trusted provider, such as Verisign or Thawte. For a full 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. Note that the Salesforce feature that validates login IP ranges continues to be in effect for single sign-on users. For more information, see "Restrict Where and When Users Can Log In To Salesforce" in the Salesforce Help.
- You may need to map your organization's internal usernames and Salesforce usernames. If your organization does not follow a standard mapping, you may be able to extend 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 do not enable single sign-on for system administrators. If your system administrators are single sign-on users and your single sign-on server has an outage, they have no way to log in to Salesforce. System administrators should always be able to log in to Salesforce so they can disable single sign-on in the event of a problem.
- We recommend that you use a Developer Edition account or a sandbox when developing a single sign-on solution before implementing it in your organization. To sign up for a free Developer Edition account, go to 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 the following best practices when implementing federated single sign-on with SAML for your organization.

- Obtain the `Salesforce Login URL` value from the Single Sign On Settings configuration page and put it in the corresponding configuration parameter of your Identity Provider (sometimes called the "Recipient URL").
- Salesforce allows a maximum of three minutes for clock skew with your IDP server; make sure your server's clock is up-to-date.
- If you are unable to log in with SAML assertion, always check the login history and note the error message. Use the SAML Assertion Validator on the Single Sign On Settings configuration page to troubleshoot.
- You need to map your organization's internal usernames and Salesforce usernames. You have two choices to do this: add a unique identifier to the `FederationIdentifier` field of each Salesforce user, or 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 User ID 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 organization preference and provide all the necessary configurations.
- Use the My Domain feature to prevent users from logging in to Salesforce directly, and gives administrators 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://<my_domain_name>.my.salesforce.com/?saml=02HKiP...`
- We recommend that you use Developer Edition account or a sandbox when testing a SAML single sign-on solution. To sign up for a free Developer Edition account, go to 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 `http://cs1.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 single sign-on configuration. The default value is `https://saml.salesforce.com`.

Single Sign-On for Portals Best Practices

Customer Portals and partner portals are not available for new organizations in the Summer '13 release or later. Use Communities instead. For more information about single sign-on and SAML for Communities, see "Configuring SAML for Communities" in Getting Started With Communities. If you continue to use portals, note the following information.

- 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 for single sign-on for portals. 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 organization.

Single Sign-On 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 for single sign-on for Sites. If only one is specified, the user receives an error.
- If all three of 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 that portal login.
- More than one portal can be used with a single organization.

ENABLE SINGLE SIGN-ON FOR PORTALS

Single sign-on allows users to access all authorized network resources without having to log in separately to each resource. You validate usernames and passwords against your corporate user database or other client application rather than having separate user passwords managed by Salesforce.

You can set up 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"

UNDERSTANDING DELEGATED AUTHENTICATION SINGLE SIGN-ON

Salesforce uses the following process for authenticating users using delegated authentication single sign-on:

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, then Salesforce does not validate the username and password. Instead, a Web services call is made to the user's organization, 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 once the process is complete.

3. The Web services call passes the `username`, `password`, and `sourceIp` to your Web service. (`sourceIp` is the IP address that originated the login request. You must create and deploy an implementation of the Web service that can be accessed by Salesforce servers.)
4. Your implementation of the Web service validates the passed information and returns either `true` or `false`.
5. If the response is `true`, then the login process continues, a new session is generated, and the user proceeds to the application. If `false` is returned, then the user is informed that his or her username and password combination is invalid.

Note: There may be a momentary delay before a user can log in after being given delegated authentication due to the time required for the user account to become available in the organization.

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"

Configuring Salesforce for Delegated Authentication

To enable delegated authentication single sign-on (SSO) for your organization:

1. Contact Salesforce to enable delegated authentication single sign-on for your organization.
2. Build your single sign-on Web service:
   a. In Salesforce, download the Web Services Description Language (WSDL) file `AuthenticationService.wsdl` from Setup by entering *`Download Delegated Authentication WSDL`* in the `Quick Find` box, then selecting **Download Delegated Authentication WSDL**. The WSDL describes the delegated authentication single sign-on service and can be used to automatically generate a server-side stub to which you can add your specific implementation. For example, in the WSDL2Java tool from Apache Axis, you can use the `--server-side` switch. In the wsdl.exe tool from .NET, you can use the `/server` switch.

   For a sample request and response, see Sample SOAP Message for Delegated Authentication on page 21.

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

   Because Salesforce does not use the `password` field other than to pass it back to you, you do not need to send a password in this field. Instead, you could pass another authentication token, such as a Kerberos Ticket so that your actual corporate passwords are not passed to or from Salesforce.

   You can configure the Salesforce delegated authentication authority to allow only tokens or to accept either tokens or passwords. If the authority only accepts tokens, a Salesforce user cannot log in to Salesforce directly, because they cannot create a valid token. However, many companies choose to allow both tokens and passwords. In this environment, a user could still log in to Salesforce through the login page.

   When the Salesforce server passes these credentials back to you in the `Authenticate` message, verify them, and the user will gain access to the application.

3. In Salesforce, specify your organization's single sign-on gateway URL from Setup by entering *`Single Sign-On`* in the `Quick Find` box, selecting **Single Sign-On Settings**, then clicking **Edit**. Enter the URL in the **Delegated Gateway URL** text box.

   For security reasons, Salesforce restricts the outbound ports you may specify to one of the following:

   - 80: This port only accepts HTTP connections.
   - 443: This port only accepts HTTPS connections.
   - 1024–66535 (inclusive): These ports accept HTTP or HTTPS connections.

4. Optionally, check the **Force Delegated Authentication Callout** box.

   Note: When this box is unchecked, a call is not made to the SSO endpoint if the login attempt first fails because of login restrictions within the Salesforce organization. If you must record every login attempt, then check this box to force a callout to the SSO endpoint regardless of login restriction failures.

5. Enable the "Is Single Sign-On Enabled" permission.

Important: If single sign-on 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. Furthermore, the single sign-on authority usually

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"

handles login lockout policies for users with the "Is Single Sign-On Enabled" permission. However, if the security token is enabled for your org, then 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 the sample code for .NET from the Salesforce Developers website.

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 in conjunction 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

This is implemented in `simple.asmx.cs`. This file declares a new class, `SimpleAdAuth`, that is a Web service with one method: `Authenticate`. There are a number of attributes declared on the method. These control the formatting of the expected request and the generated response, and 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 not valid.

Sample 2

This is a more complex example that 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 class `SingleSignOn` that can generate an authentication token and implements the authentication service to later verify that token. The generated token consists of a token number, expiry 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 and/or linked to from an intranet site. This 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 lets the user into Salesforce. `intranet.aspx` is a simple page that links to `gotosfdc.aspx` so you can see this in action.

FREQUENTLY ASKED QUESTIONS

How do I enable single sign-on?

Salesforce offers the following ways to use single sign-on:

- Federated authentication using Security Assertion Markup Language (SAML) allows you to send authentication and authorization data between affiliated but unrelated Web services. This enables you to sign on to Salesforce from a client application. Federated authentication using SAML is enabled by default for your organization.
- Delegated authentication single sign-on enables you to integrate Salesforce with an authentication method that you choose. This enables you to integrate authentication with your LDAP (Lightweight Directory Access Protocol) server, or perform single sign-on by authenticating using a token instead of a password. You manage delegated authentication at the permission level, allowing some users to use delegated authentication, while other users continue to use their Salesforce-managed password. Delegated authentication is set by permissions, not by organization.

  The primary reasons for using delegated authentication include:

  - Using a stronger type of user authentication, such as integration with a secure identity provider
  - Making your login page private and accessible only behind a corporate firewall
  - Differentiating your organization from all other companies that use Salesforce in order to reduce phishing attacks

  You must request that this feature be enabled by Salesforce. Contact Salesforce to enable delegated authentication single sign-on for your organization.
- Authentication providers let your users log in to your Salesforce organization using their login credentials from an external service provider. Salesforce supports the OpenId Connect protocol that allows users to log in from any OpenID provider such as Google, PayPal, LinkedIn and other services supporting OpenID Connect. When authentication providers are enabled, Salesforce does not 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://na1.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 13.

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

D

Delegated authentication
- configuring single sign-on 46
- sample implementations 48
- single sign-on 45

E

Error page
- customizing in SAML 13

I

Identity provider
- values 9

J

Just-in-time provisioning
- example SAML assertions 14

Just-in-Time provisioning
- community requirements 35
- portal requirements 32
- requirements 30

Just-in-Time provisioning errors 38

L

Logging in
- SAML start page 13

Logging out
- SAML 13

P

Portals
- single sign-on 44

S

SAML
- about 3
- custom error page 13

SAML *(continued)*
- example assertions 14
- Just-in-Time for communities 35
- Just-in-Time for portals 32
- Just-in-Time provisioning 30
- Just-in-Time provisioning errors 38
- Just-in-Time provisioning requirements 30
- login history 29
- login page 13
- logout page 13
- prerequisites 8
- single sign-on 4
- start page 13
- validating single sign-on 26
- validation errors 27
- viewing single sign-on 25

Security
- Just-in-Time for communities 35
- Just-in-Time for portals 32
- Just-in-Time provisioning 30
- Just-in-Time provisioning requirements 30
- portals single sign-on 44

Single sign-on
- best practices 41
- configuring delegated authentication 46
- debugging 26
- delegated authentication 45, 48
- example SAML assertions 14
- FAQ 49
- identity provider values 9
- login history 29
- overview 1
- portals 44
- prerequisites 8
- SAML 4
- SAML validation 26
- viewing 25