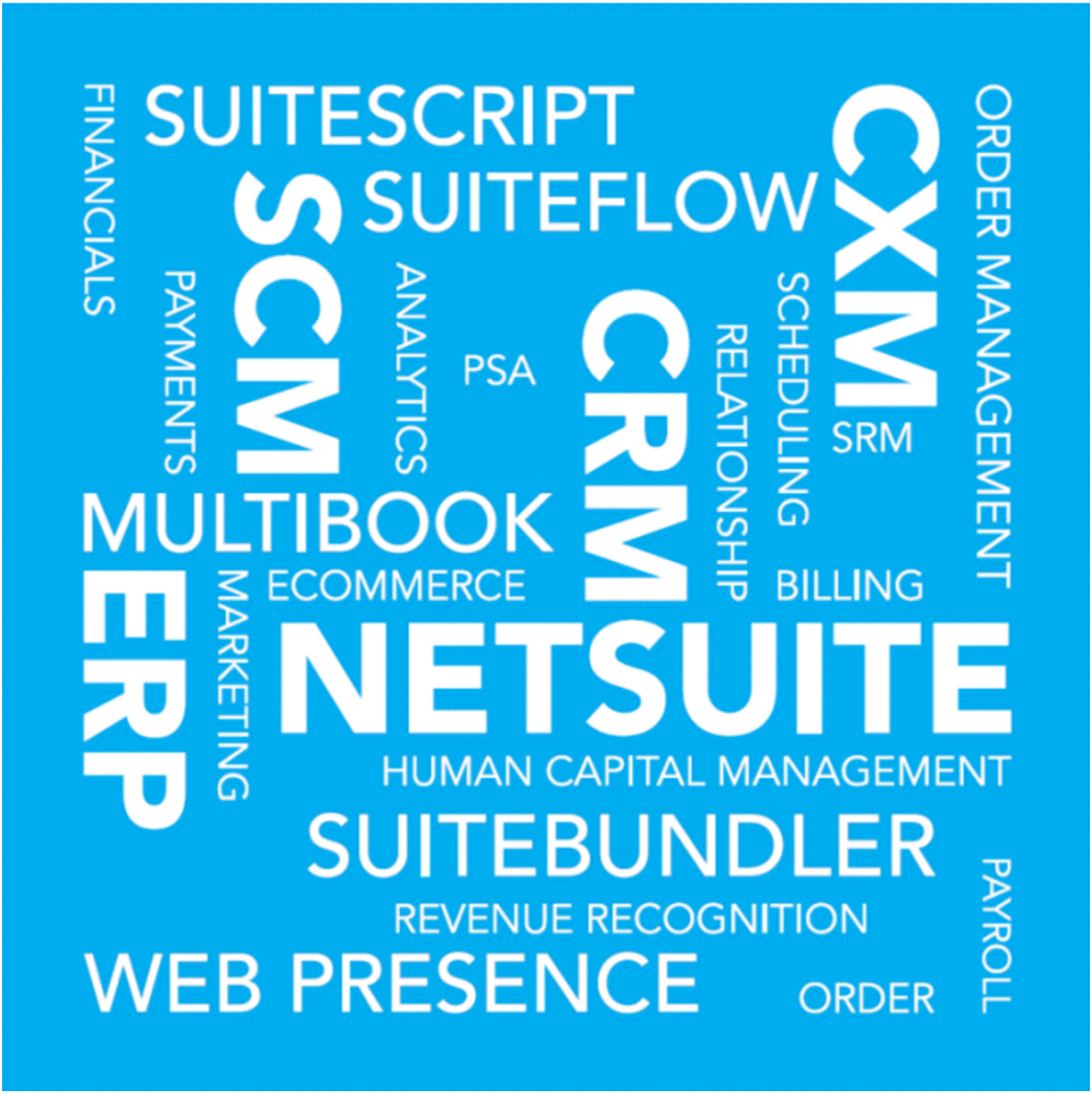
Inbound Single Sign-on Guide



April 11, 2018 2018.1

and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and "as available", for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer's use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle's other customers, Oracle may slow down or throttle Customer's excess use until such time that Customer's use stays within reasonable limits. If Customer's particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer's actual usage pattern.

Beta Features

Oracle may make available to Customer certain features that are labeled "beta" that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party's property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle's standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle's customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a "beta" label.

Table of Contents

Inbound Single Sign-on

The inbound single sign-on feature allows users to go directly from an external user-authenticating application to NetSuite, without having to log in separately to NetSuite. This feature allows a one-way trust relationship to be established between the external application and NetSuite, so that after users present login credentials to the external application, they can gain access to NetSuite as well.

Inbound single sign-on access generally has two types of users:

- Customers, who want to integrate their own NetSuite data with an external application's data.
- Application providers, who want to integrate customer data stored in their data center with their customers' NetSuite data.

With inbound single sign-on, authentication information from the external application is passed to NetSuite through an encrypted token, and a dynamically constructed URL redirects users from the external site to a NetSuite landing page. A mapping between each user's external credentials and their NetSuite credentials is created, either through a web services operation or through the user's login to NetSuite on their first single sign-on access.

To implement inbound single sign-on from your site to NetSuite, you must set up a trust relationship, with OpenSSL encryption keys, between your application and NetSuite. These keys are used to produce and interpret encrypted tokens. You also must write application code that dynamically constructs the redirect URL for each inbound single sign-on user. HTTP POST requests are not supported. NetSuite provides a downloadable kit with tools you can use for these tasks.

To get started with inbound single sign-on:

1. Review this guide, including the following, to ensure that the NetSuite inbound single sign-on feature will meet your needs.
   - For an overview of how inbound single sign-on works, see Understanding Inbound Single Sign-on.
   - For instructions for setting up and implementing an inbound single sign-on integration, see Setting Up Inbound Single Sign-on. This section includes the following information:
     - Initial Setup for the Inbound Single Sign-on Feature
     - Implementing Inbound Single Sign-on in an External Application
     - Generating Keys Using OpenSSL
     - Creating the Initial Mapping of the Administrator Role for Inbound Single Sign-on
     - Creating Single Sign-on Code Using SSOUrl
   - For instructions for setting up inbound signle sign-on mappings, see Mapping Users and Roles for Inbound Single Sign-on Access to NetSuite
   - For technical background, see Technical Summary of Inbound Single Sign-on.
2. Be aware of the following:
   - Inbound single sign-on access is supported for the NetSuite application, including the Customer Center, and for NetSuite web stores.
   - Inbound single sign-on access to the Customer Center is supported for NetSuite users classified as customers and for customer contacts.
   - Inbound single sign-on access to NetSuite respects IP address restriction rules. For information about this feature, see the help topic Enabling and Creating IP Address Rules.
   - Inbound single sign-on access to web store is supported for custom checkout domains, multi-site implementations, and sites customized with SSP applications. Access is also supported

for Reference Cart & One Page Checkout, the NetSuite reference implementation of the web store checkout process. See the help topic Inbound Single Sign-on Access to Web Store.

3. After you have confirmed that you want to implement inbound single sign-on, contact your account manager to purchase the feature.

Alternate Inbound Single Sign-on Mechanisms

NetSuite supports two other features that do not use this NetSuite version of token-based inbound single sign-on authentication:

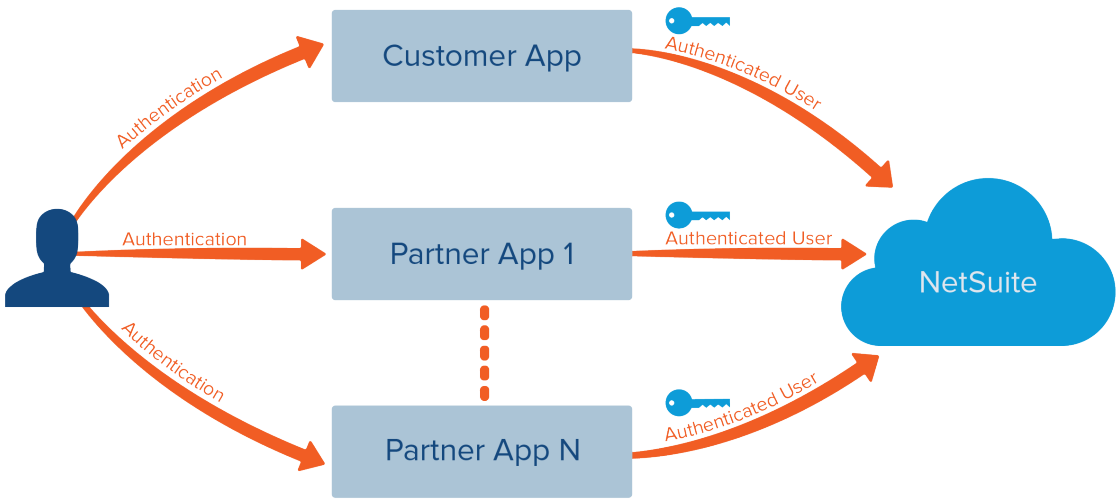
- The SAML Single Sign-on feature supports inbound single sign-on access to NetSuite using authentication from a SAML v2.0-compliant third-party identity provider. See the help topic SAML Single Sign-on.

  Note: It is not necessary to purchase the NetSuite Inbound Single Sign-on feature if you want to implement SAML Single Sign-on in NetSuite.

- The OpenID Single Sign-on feature supports inbound single sign-on access from Google Apps to NetSuite, relying on Google Accounts as the trusted system of authentication. See the help topic OpenID Single Sign-on.

Inbound Single Sign-on Overview

The NetSuite inbound single sign-on feature enables users to move directly from an external user-authenticating web application to NetSuite without additional authentication. This feature provides token-based integration.

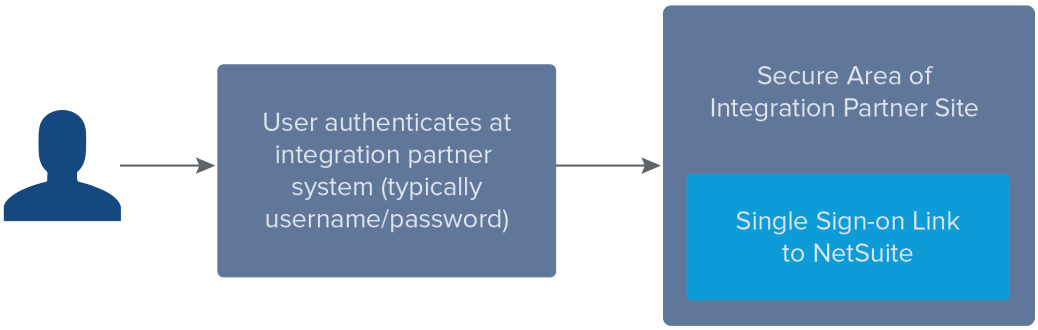


The external application uses an encrypted token to pass the user's identity to NetSuite, NetSuite verifies the token, then logs in the user. This single sign-on mechanism can be implemented through a link in the external application user interface, or through web services calls that use the ssoLogin operation.

Understanding Inbound Single Sign-on

The following steps outline how inbound single sign-on to NetSuite works.

1. In most cases, a user initiates inbound single sign-on access to NetSuite by clicking a link in an authenticated area of an external site. This site can be used in conjunction with either the NetSuite application user interface or a NetSuite web store.



   Alternatively, the web services ssoLogin operation can be used to initiate inbound single sign-on access programmatically.

2. When a user initiates inbound single sign-on access, the external application produces a token that includes the following information:
   - the user's external application company ID

     This value is a string used by the external application to determine the company with which a user is associated, for example `ABCAutoParts`. It cannot contain spaces.
   - the user's external application user ID

     This value is a string used by the external application as a user identifier, for example `John.Smith`. It cannot contain spaces.
   - the current timestamp

     The timestamp string is a decimal representation of the number of milliseconds since January 1, 1970, 00:00:00 GMT.

   For more information about the token, see the following topics:
   - Tables of Single Sign-on Redirect URL Parameters
   - Elements of the Authentication Token String
   - Example Inbound Single Sign-on Token
3. The external application encrypts the information included in the token.
   - To encrypt the token, the external application must have access to a private key generated using OpenSSL. To interpret the encrypted token, NetSuite must have access to a public key extracted from this private key.
   - The inbound single sign-on kit includes Java classes you can use to produce the private and public keys. For instructions, see Generating Keys Using OpenSSL.
4. After encryption of the token, the external application causes the user's browser to perform a redirect to NetSuite. HTTP POST requests are not supported.
   - The redirect is either to the NetSuite application or to the web store, based on the target set in the code (either `app` or `site`).
   - The redirect uses a URL constructed specifically for this inbound single sign-on access. This URL includes required parameters such as:
     - a hex-encoded, encrypted string representing the token
     - the unique partner ID assigned by NetSuite Customer Support
     - For NetSuite application access only, the remote company ID, which is the company ID used in the token

- For web store access only, the domain, NetSuite company ID, and site ID

> **Note:** For more information on required and optional parameters, see Tables of Single Sign-on Redirect URL Parameters.

- This URL is valid for up to 15 minutes after the timestamp included in the encrypted token.
- The inbound single sign-on kit includes a Java class you can use to create code that dynamically constructs this URL. For instructions, see Creating Single Sign-on Code Using SSOUrl.

5. NetSuite receives the token. Based on a unique partner ID assigned by NetSuite when inbound single sign-on is set up, NetSuite determines the public key that should be used to decrypt the token. After the token is decrypted, if the timestamp is valid, the request is honored.
6. NetSuite checks for a user mapping between the external application and NetSuite.
   - If there is an existing mapping, the user is logged in as defined by the mapping.
   - If a mapping does not exist, the user is prompted to provide NetSuite credentials to create the mapping.

   > **Important:** A user with an Administrator role must create the initial mapping from the external application to NetSuite. For more information, see Creating the Initial Mapping of the Administrator Role for Inbound Single Sign-on.

   After the initial mapping to the Administrator role is completed:

   - For web store access, the administrator is required to use the web services mapSso operation to create the account mapping for multiple users so that it is available before users initiate single sign-on access.
   - For NetSuite access, the administrator can use the web services mapSso operation to create the account mapping for multiple users, or can instruct users to create their own mappings. See Mapping Users and Roles for Inbound Single Sign-on Access to NetSuite.
7. After the user's identity has been verified, a NetSuite landing page displays.
   - The default landing page for NetSuite application access is the user's home page.
   - The default landing page for web store access is the site home page.
   - A non-default landing page can be defined by adding a `landingurl` parameter to the redirect URL.
   - If no mapped NetSuite identity is found, the NetSuite inbound single sign-on login page displays.
8. A NetSuite session initiated through inbound single sign-on is subject to standard NetSuite session timeout rules.
   - By default, the user is redirected to an inbound single sign-in login page on session timeout or error.
   - This NetSuite login page can be hidden by setting the redirect URL's `hideloginpage` parameter to true, so that the user is returned to a different page, such as an external application page. In this case, a `returnurl` parameter also must be added, to specify this alternate page.
9. The user can log out from an inbound single sign-on session in the same manner as any other NetSuite session.
   - By default, the user is redirected to the inbound single sign-in login page on logout.

- If the redirect URL includes the `returnurl` parameter, the user is redirected to the page specified by this parameter instead. It is not necessary to set the `hideloginpage` parameter to `T` (true) to vary the page on logout.

Setting Up Inbound Single Sign-on

See the following two procedures for important information:

1. Initial Setup for the Inbound Single Sign-on Feature

   This required procedure outlines requirements for setting up the inbound single sign-on feature in your account. It is guided by NetSuite Customer Support, and occurs after you have contacted your NetSuite account representative and purchased the Inbound Single Sign-on feature.

2. Implementing Inbound Single Sign-on in an External Application

   This procedure outlines options for inbound single sign-on integration from an external application to NetSuite.

Note: It is not necessary to purchase the NetSuite Inbound Single Sign-on feature if you want to implement SAML Single Sign-on in NetSuite. For more information, see the help topic SAML Single Sign-on. See also Alternate Inbound Single Sign-on Mechanisms.

Initial Setup for the Inbound Single Sign-on Feature

Warning: You must contact your NetSuite account representative and purchase the inbound single sign-on feature to properly initiate the setup process. Do not attempt to complete these steps on your own. Wait until you are contacted by NetSuite Customer Support to begin the initial setup of this feature.

To complete the initial setup of the inbound single sign-on feature in your account:

1. Contact your account representative to purchase the inbound single sign-on feature.
   a. NetSuite Customer Support will open a new support case, and contact you for specific information.
   b. NetSuite Customer Support will ask you to generate a public and private key pair using OpenSSL. See Generating Keys Using OpenSSL.
   c. NetSuite Customer Support will ask you to provide the generated public key through the support case.
   d. NetSuite Customer Support will associate the public key with an Inbound Single Sign-on Partner ID, and will provide this unique Partner ID to you.
2. After NetSuite Customer Support guides you through the initial setup, and provides you with your unique Partner ID, you will be ready to implement inbound single sign-on in your application. See Implementing Inbound Single Sign-on in an External Application.

Implementing Inbound Single Sign-on in an External Application

Before you attempt to implement any of the following options in your external application, you must have already contacted your NetSuite account administrator and purchased the Inbound Single Sign-on

feature. Also, NetSuite Customer Support must have already guided you through the steps outlined in Initial Setup for the Inbound Single Sign-on Feature.

To implement inbound single sign-on in an external application:

You can choose any of the following options to implement inbound single sign-on from an external application to NetSuite:

1. Download the kit for implementing inbound single sign-on:

   https://system.netsuite.com/download/NLSingleSignOn.zip.

   > **Note:** A checksum file is also available: https://system.netsuite.com/download/NLSingleSignOn.sha512.

2. Add the `ssov3.jar` file from this kit to your Java classpath.

   You need the contents of this .jar file to facilitate compilation of your single sign-on integration code and to generate keys for token encryption.

   > **Note:** Java developers can add the ssov3.jar to your classpath, along with the Java run-time environment classes. Source code is also provided for developers in non-Java environments as a template for implementation.

3. Write application code that dynamically constructs redirect URLs to be used when users initiate inbound single sign-on access. HTTP POST requests are not supported. See Creating Single Sign-on Code Using SSOUrl.
4. Write web services code for the single sign-on integration as needed.

   You can programmatically initiate access with ssoLogin, and/or programmatically map users' external credentials to NetSuite credentials. See SuiteTalk (Web Services) Single Sign-on Operations.
5. Provide error handling for status codes returned from NetSuite inbound single sign-on sessions. See Error Handling for Inbound Single Sign-on.
6. To prevent single sign-on users from directly logging in to NetSuite, create a custom role that is designated as **Single Sign-on Only**, and assign this role to single sign-on users. See Setting Up a Single Sign-on Only Role.

Generating Keys Using OpenSSL

As described in the section Initial Setup for the Inbound Single Sign-on Feature, NetSuite Customer Support will ask you to generate a public and private key pair using OpenSSL. The public key is provided to NetSuite, for use in creating your unique Partner ID. You will use the private key in your implementation to encrypt authentication tokens.

To generate keys for inbound single sign-on:

1. Either append the `openssl` subdirectory provided in the inbound single sign-on kit to your PATH, or download source code from http://www.openssl.org/source.

   The binaries included in the `openssl` directory are derived from `openssl0.9.6.tar.gz`. If you are creating your own binaries from a downloaded source package, follow directions in the **INSTALL** file appropriate to your operating system.
2. After `openssl` is installed and in your PATH, type **openssl** to get the following prompt:

   ```
   OpenSSL>
   ```

3. At the prompt, use the following command to generate a private key:

```
OpenSSL> genrsa -out <privKey.pem> -rand <f1><s><f2><s><f3><s><f4><s><f5> 2048
```

- <privKey.pem> is the desired name of the output file.
- <f1> through <f5> are names of files used as random seeds.
- <s> is a separator:
  - ; for Windows
  - , for OpenVMS
  - : for all other operating systems
- This process generates a private key with a modulus length of 2048 bits. The output file format produced (PEM) is not appropriate for use by NetSuite tools, however, and must be properly formatted, as described in the following step.

4. Convert the private key to DER using the Pem2Der class provided by NetSuite, by typing the following Java command:

```
java com.netledger.forpartners.encryption.Pem2Der <privKey.pem>
<privKey.der>
```

5. Extract the public key from the private key using the Priv2Pub class provided by NetSuite, by typing the following command:

```
java com.netledger.forpartners.encryption.Priv2Pub
<privKey.der> <pubKey.der>
```

- <privKey.der> and <pubKey.der> of this last command are your public and private keys.

Note: As described in the section Initial Setup for the Inbound Single Sign-on Feature, you will provide the public key to NetSuite Customer Support through the support case. Maintain the private key for use by the NetSuite SSOUrl class. See Creating Single Sign-on Code Using SSOUrl.

Creating the Initial Mapping of the Administrator Role for Inbound Single Sign-on

The initial single sign-on account mapping must be a mapping to an Administrator role in NetSuite. This Administrator role mapping serves as authorization that NetSuite trusts the external authentication system. This requirement gives NetSuite administrators control over when single sign-on functionality is available to their users.

Note: The following procedure assumes that you have completed all tasks as described in the Initial Setup for the Inbound Single Sign-on Feature and Generating Keys Using OpenSSL topics. (You have added the ssov3.jar file from the inbound single sign-on kit to your Java classpath, NetSuite Customer Support has assisted with the generation of the public and private keys, and NetSuite Customer Support has provided you with the Partner ID for this account.)

A user with an Administrator role in this account must create the initial mapping between NetSuite and the external authentication system. To generate the token necessary to complete the initial mapping, you can use the web services mapSso operation, or the external third-party application if it is available. You can also use the tool included in the Inbound Single Sign-on kit to generate the URL, which includes the token necessary for this procedure.

To create a token with the ssov3.jar file:

1. Call the `ssov3.jar` file.
2. Specify the appropriate parameters:
   a. `rc` = the target account number.
   b. `p` = the Partner ID assigned byNetSuite Customer Support.
   c. `ru` = the Administrator role to be used for the initial mapping.
   d. `t` = the type of URL you want to generate, either `app` (for inbound access to the NetSuite UI) or `site` (for inbound access to your website).

   See Tables of Single Sign-on Redirect URL Parameters for more information.
3. The tool generates a URL with a token in the form: `<domain>/app/login/secure/sso.nl<partnerID><TokenBigLongString>`

   Note: The token is valid for 15 minutes. You must complete the following steps before the token expires, or generate a new token.
4. Copy and paste the URL into your browser's address bar and click **Enter**. The NetSuite Partner Login page displays.
5. On the NetSuite Partner Login page, enter your NetSuite email address and password.
6. Click **Log in**. The NetSuite Choose a role to create the mapping page displays.
7. Click your Administrator role for this account.

   Note: If you have Administrator roles in more than one account, ensure you are selecting the correct Administrator role for this specific account.

   The initial mapping of the Administrator role is now complete.

After the initial mapping is completed, other users and roles can now be mapped to the external application. The web services mapSso operation can be used to create the account mappings for multiple users so that they are available before users initiate single sign-on access. (This method of mapping is required for web store access.)

Creating Single Sign-on Code Using SSOUrl

The quickest way to create inbound single sign-on code is to use the SSOUrl Java class provided in the downloadable kit. This class is available to you after you have downloaded the kit and added the `ssov3.jar` file to your Java classpath.

The `SSOUrl.java` file provides a template for Java code, along with explanatory comments. You can use this file to guide your creation of single sign-on integration code in Java. A command-line utility that you can run from a shell is also provided as an alternative.

Note: If you are NOT using the NetSuite SSOUrl class to generate redirect URLs, then you will need to construct them using your own methods from the base elements described in `SSOUrl.java`.

If you are using the SSOUrl class to implement inbound single sign-on integration, your application code needs to do the following:

- Initialize the SSOUrl class with the file name of the private key used to encrypt the authentication token.
- Set the target of the inbound single sign-on access to either the NetSuite application (app) or the web store (site), so that the base URL for the integration is correctly generated:

- for the NetSuite application:

```
https://system.netsuite.com/app/login/secure/sso.nl
```

  HTTP POST requests are not supported.
- for the web store:

```
https://checkout.mycompanystore.com/app/site/backend/sitesso.nl
https://checkout.netsuite.com/app/site/backend/sitesso.nl
https://checkout.na1.netsuite.com/app/site/backend/sitesso.nl
https://checkout.na2.netsuite.com/app/site/backend/sitesso.nl
```

- For inbound single sign-on access to web store, set the domain for your web store.
  - If you have a custom checkout domain, set the domain appropriately.
  - If you are using a NetSuite-hosted checkout domain, set it to the appropriate checkout domain for your data center. The NetSuite company ID and site ID URL parameters are also required for web store access. For more information, see Web Store Access Only Parameters.
- Provide the single sign-on link as a link to an internal page that uses the `SSOUrl.getURL(companyId, userId)` method to dynamically construct a redirect URL to a landing page in the NetSuite application or web store. This URL should include all required parameters and any desired optional parameters. HTTP POST requests are not supported.
- Redirect the browser to the constructed URL.

For more information, see the following:

- Tables of Single Sign-on Redirect URL Parameters
- Example Single Sign-on Token and Redirect URLs

Tables of Single Sign-on Redirect URL Parameters

The following tables describe the parameters used in single sign-on redirect URLs. HTTP POST requests are not supported.

Review all of the tables to ensure that you are including all of the parameters required for your purposes.

- Parameters used for both Application and Web Store Access
- NetSuite Application Access Only Parameters
- Web Store Access Only Parameters

In addition, see the following:

- For further insight into parameter usage, review the contents of the `SSOUrl.java` file.
- For example redirect URLs, see Example Single Sign-on Token and Redirect URLs.

Parameters used for both Application and Web Store Access

The following table describes the parameters used for both the NetSuite application access (`app`) and web store access (`site`).

Name	Description	Required/Optional	Programmatic Parameter	Command-Line Parameter
authentication token	string representing the encrypted token	Required	a	

Name	Description	Required/Optional	Programmatic Parameter	Command-Line Parameter
	For more information, see Elements of the Authentication Token String. See also Example Inbound Single Sign-on Token.			
partner ID	unique ID assigned by NetSuite for use with inbound single sign-on this ID is associated with the public key you provided to NetSuite	Required	pid	p
landingurl	page that first displays for inbound single sign-on access, other than default Defaults are: ■ for NetSuite application access, the user's home page ■ for web store access, the site home page	Optional	landingurl	l
hideloginpage	Boolean indicating whether to hide the default inbound single sign-in login page from users and instead go to the page specified by the returnurl parameter By default, set to false. NetSuite recommends that it be set to true for web store access.	Optional	hideloginpage	h
returnurl	page where single sign-on users are redirected on session logout, timeout, and errors Default is the inbound single sign-on login page.	Required if hideloginpage set to true	returnurl	r

Elements of the Authentication Token String

The format of the authentication token string prior to encryption is:

```
<companyID><space><userID><space><timestamp>
```

The `companyID` and `userID` elements represent the credentials used by the external application. (These credentials will be mapped to the NetSuite identity during inbound single sign-on.) Because spaces are used to delimit subtokens, `companyID` and `userID` elements cannot contain spaces.

See the following information about each element in the string:

- The `companyID` element is used by the external application to determine the company with which a user is associated, for example, `ABCAutoParts`. The `companyID` that you should use can vary. The goal is to ensure that the application token string is unique.
  - If you are a partner building an application for another company, the `companyID` should be a unique identifier of that company. You could use the company's name, or any identifier you use to identify that company.

  - If you are building an integration for your own company, use your company name.
  - In any case, you can always use the NetSuite account ID as the `companyID`. To locate the account ID, go to Setup > Company > Setup Tasks > Company Information. The account ID field is located near the bottom of the right column.
- The `userID` string used by the external application as a user identifier, for example, `John.Smith`. It cannot contain spaces.
- The `timestamp` string is a decimal representation of the number of milliseconds since January 1, 1970, 00:00:00 GMT. The token is valid for 15 minutes after the timestamp contained in it.

NetSuite Application Access Only Parameters

The following table describes the NetSuite application access only parameters.

Name	Description	Required/Optional	Programmatic Parameter	Command-Line Parameter
partner account / remote company ID	External application-assigned ID for your company. This value is identical to the `companyID` value used in the token. For more information about the `companyID`, see Elements of the Authentication Token String.	Required if target is app	pacct	rc
partner user ID / remote user ID	External application-assigned ID for your user. This value is identical to the userID value used in the token.	Required if target is app	puid	ru
application domain	Allows specification of domain and data center, for example, system.na1.netsuite.com. If specified, the application domain for the base URL will be overridden with provided value.	Optional		ad

Web Store Access Only Parameters

The following table describes the web store only access parameters used in single sign-on redirect URLs. These parameters are used when the target is `site`. HTTP POST requests are not supported.

Important: Always specify the NetSuite-hosted checkout domains (including the correct data center) in addition to the `c` parameter for faster routing if the performance of the login operation is a concern.

Name	Description	Required/Optional	Programmatic Parameter	Command-Line Parameter
domain	Your custom checkout domain, for example: `checkout.mycompany.com`. If you use a NetSuite-hosted checkout domain, enter the domain. The domain must include the correct data center identifier.	Required for custom checkout domains. Recommended for NetSuite-hosted checkout domains.		d

Name	Description	Required/Optional	Programmatic Parameter	Command-Line Parameter
	See Checkout Domains for Data Centers.			
NetSuite company ID	NetSuite-assigned account ID for your company	Required for NetSuite-hosted checkout domains.	c	c
site ID	internal ID on a NetSuite website record; distinguishes among multiple sites in your web store integer displayed on the Web Site Preview page, as shown in Finding the Site ID Parameter A site ID of **1** is valid even with only a single site.	Required when the c parameter is used.	n	s

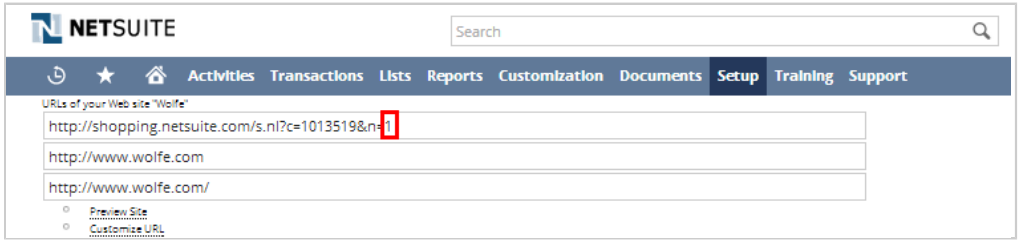
Checkout Domains for Data Centers

If you use a NetSuite-hosted checkout domain, for the domain parameter, use the correct data center identifier in your URL.

- NA West: `checkout.netsuite.com`
- NA East: `checkout.na1.netsuite.com`
- NA Northwest: `checkout.na2.netsuite.com`
- NA Central: `checkout.na3.netsuite.com`
- EU West: `checkout.eu1.netsuite.com`
- EU Central: `checkout.eu2.netsuite.com`

Finding the Site ID Parameter

You can find the value of the site ID parameter to set for a multi-site environment at Setup > Site Builder > Preview Web Site:



Note: A site ID of 1 is valid for a single site as well as for a multi-site environment.

Example Single Sign-on Token and Redirect URLs

Review the following examples to get a better understanding of inbound single sign-on tokens and redirect URLs:

- Example Inbound Single Sign-on Token

- Example Redirect URL for the NetSuite Application
- Example Redirect URL for the Web Store
- Example Redirect URL for Intermediate Third-party Login to Web Store

For details about the parameters used in redirect URLs, see Tables of Single Sign-on Redirect URL Parameters.

Example Inbound Single Sign-on Token

The following example illustrates the three stages of generating an inbound single sign-on token. The token is created with:

- the external application-assigned remote company ID, and
- the remote user ID for the user, and
- the current timestamp.

The token is then encrypted using a private key, and then hex-encoded so it can be passed as a redirect URL parameter.

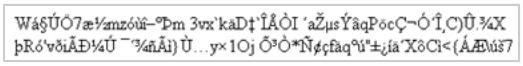
Note: The hex-encoded, encrypted token string that is used as the URL parameter.

Stages of Token Generation

1. **Plain Text String:**

   ```
   ABCAutoParts John.Smith 1225479286770
   ```

2. **Encrypted String:**

3. **Hex-Encoded, Encrypted String:**

   ```
   57E1A7DA1CD637E6BD1F6D7AF3F9EE96B0DE6D1E0D337678606BE4448760CEC5D249031CA0B4618EB50C731703DDE27
   150F663C7AC11D3B4CEB84329DB2EBE58FE1D16520FF3271476F069C31DD0BCDA0AAFB4BEF1C3EC0F7DD98579D7314F
   6A0AD5B3D22AD1A2E766E471B0FA22B1BFEDE4Br58Frr315EC3C7BC1C65CFA9A37
   ```

Example Redirect URL for the NetSuite Application

The following is an example redirect URL for inbound single sign-on access to the NetSuite application:

```
https://system.netsuite.com/app/login/secure/sso.nl?pid=198765&pacct=ABCAutoParts&puid=John.Smi
th&a=57E1A7DA1CD637E6BD1F6D7AF3F9EE96B0DE6D1E0D337678606BE4448760CEC5D249031CA0B4618EB50C731703
DDE27150F663C7AC11D3B4CEB84329DB2EBE58FE1D16520FF3271476F069C31DD0BCDA0AAFB4BEF1C3EC0F7DD98579D
7314F6A0AD5B3D22AD1A2E766E471B0FA22B1BFEDE4Br58Frr315EC3C7BC1C65CFA9A37
```

The base URL in the redirect URL is determined by the target set in integration code; in this case, the target is set to `app`.

Values for the URL parameters in this example also are set in integration code: the partner ID (`pid`), the remote company id (`pacct`), the remote user ID (`puid`), and the hex-encoded encrypted token string (`a`).

Important: The parameters listed above are valid parameters for this use case. For more information, see the Tables of Single Sign-on Redirect URL Parameters.

Do not use the `ck` and `cktime` parameters described in the Example Redirect URL for Intermediate Third-party Login to Web Store.

Example Redirect URL for the Web Store

The following is an example redirect URL for inbound single sign-on access to the web store:

```
https://checkout.netsuite.com/app/site/backend/sitesso.nl?a=57E1A7DA1CD637E6BD1F6D7AF3F9EE96B0D
E6D1E0D337678606BE4448760CEC5D249031CA0B4618EB50C731703DDE27150F663C7AC11D3B4CEB84329DB2EBE58FE
1D16520FF3271476F069C31DD0BCDA0AAFB4BEF1C3EC0F7DD98579D7314F6A0AD5B3D22AD1A2E766E471B0FA22B1BFE
DE4Br58Frr315EC3C7BC1C65CFA9A37&pid=198765&hideloginpage=T&returnurl=http://www.abcautoparts.co
m/&c=198765&n=1
```

The base URL in the redirect URL is determined by the target set in integration code; in this case, the target is set to `site`.

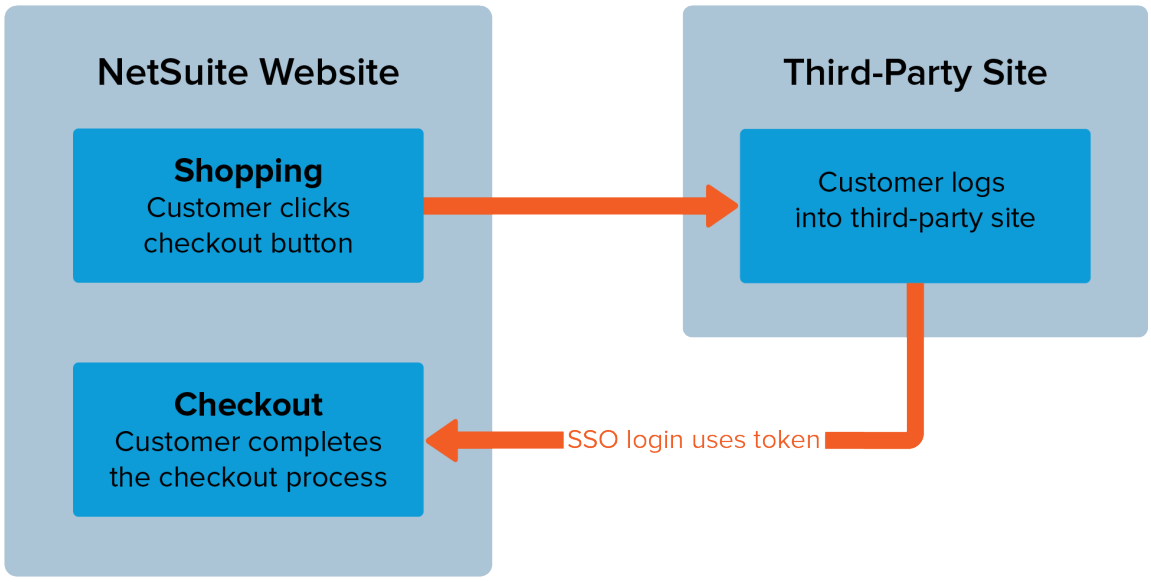
Values for the URL parameters in this example also are set in integration code: the hex-encoded encrypted token string (`a`), the partner ID (`pid`), the hide login page indicator (`hideloginpage`), the return URL (`returnurl`), the NetSuite-assigned company ID (`c`), and the site id (`n`).

Important: The parameters listed above are valid parameters for this use case. For more information, see the Tables of Single Sign-on Redirect URL Parameters.

Do not use the `ck` and `cktime` parameters described in the Example Redirect URL for Intermediate Third-party Login to Web Store.

Example Redirect URL for Intermediate Third-party Login to Web Store

Inbound single sign-on supports the workflow where a customer visits a NetSuite shopping site, adds an item to the cart, clicks the Checkout button, and then is directed to a third-party site for login. As shown in the diagram below, after logging into the third-party site, the customer is directed to NetSuite checkout servers to complete a transaction.



To ensure that the shopping cart contents persist to the NetSuite checkout servers, parameters that allow the checkout servers to determine the original session must be included in the single sign-on call to NetSuite.

Important: The `ck` and `cktime` parameters described in the following procedure should only be used in situations when there is an intermediate third-party login required before proceeding to the Web Store.

To ensure synchronization between the NetSuite web store checkout server and the shopping server:

1. Include two additional parameters, `ck` and `cktime`, in the customized checkout link pointing to third-party servers for login. You can include these parameters by using tags in the customization text.

   You might, for example, put the tags directly into the URL you are substituting for the checkout URL as:

   ```
   &ck=< _NLSHOPPERID_>&cktime=< _NLCOOKIETIMESTAMP_>
   ```

2. Upon receiving these parameters on the third-party login resource, read them, and then save them for addition to the URL to which the customer is redirected for single sign-on after login at the third-party site.

   Example:

   ```
   https://checkout.netsuite.com/app/site/backend/sitesso.nl?landingurl=http%3A%2F%2Fshopping.f.ne
   tsuite.com%2Fs.nl%3Fc%3D1035737&pid=1&c=1035737&a=792C4B61EF9BE695E9E9375FD78D24F25200EDEEF01A4
   16B03A2AAC41EE0E2C31F4503D33F0E7FED1C154BFD559B7AC9D8E1B9DE4B9882D4FF9488DB11867BCE03B1A91C9388
   1B09F1FB99B0837BA0642CB58EA8B9839308503DF3ADDE3DD3F22ED37704D7C30171871C6439E0F69BCA49C6DAA2B5B
   1D2651490B6FA4E3FA4BB&ck
   =rBDDSzm-AdboaPPb&cktime=114233
   ```

The base URL in the redirect URL is determined by the target set in integration code; in this case, the target is set to `site`.

Values for the URL parameters in this example also are set in integration code: the page that first displays for inbound single sign-on access (`landingurl`), the partner ID (`pid`), the NetSuite-assigned company ID (`c`), the hex-encoded encrypted token string (`a`), the shopperid (`ck`), and a time stamp (`cktime`).

Important: The parameters listed above are valid parameters for this use case. For more information, see the Tables of Single Sign-on Redirect URL Parameters. The `ck` and `cktime` parameters are additional parameters valid for this use case only.

SuiteTalk (Web Services) Single Sign-on Operations

The web services `mapSso` operation provides the ability to automate the mapping between users' external application credentials and NetSuite credentials.

- This operation provides inbound single sign-on access to NetSuite without users having to log in to NetSuite the first time this access occurs. Instead of the mapping between their external application credentials and NetSuite credentials being created at the time of this login, the mapping is created through web services.

- Use of this operation is required for inbound single sign-on access to the web store.
- This operation is applicable to accounts that have inbound single sign-on set up, and that have access to the associated external application credentials and encryption keys needed to generate the token.
- For more information, see the web services mapSso topic, which includes code samples.

Single sign-on mappings are not copied from a production account to a sandbox account when the sandbox is refreshed. These mappings must be recreated in the sandbox account for any users who require inbound single sign-on access to that account.

The web services `login` operation provides a mechanism for external applications to automate inbound single sign-on user login to NetSuite without the user's NetSuite credentials going through the external servers.

- This operation provides inbound single sign-on access to NetSuite without users having to click a link in the external application. The activities that occur when a user clicks this type of link instead occur behind the scenes.
- This operation is applicable to users who authenticate to NetSuite through the web services `login` operation; it is not applicable to users who authenticate to NetSuite by providing user credentials in the header of their SOAP requests.
- For more information, see the web services ssoLogin topic, which includes code samples.

Important: NetSuite hosts customer accounts in multiple data centers. For that reason, the correct URL for web services access varies depending on the data center hosting the account. Your integration must incorporate logic that dynamically determines the correct URL. With the 2012.2 and later endpoints, you should use the getDataCenterUrls operation to dynamically discover the correct URL. With older endpoints, you should use the REST roles service. For details, see the help topic Using the REST roles Service to Get User Accounts, Roles, and Domains.

Error Handling for Inbound Single Sign-on

When an inbound single sign-on session is interrupted, NetSuite sends status codes to the external application. The external site can receive each of these status codes and display an appropriate error to the user.

- The following status codes may be returned if the `hideloginpage` parameter is set to T (true): (If `hideloginpage` is set to F (false), the user is redirected to the login page.)
  - `LOGIN_ERR_NO_MAPPING` - No SSO mapping of user authentication exists in NetSuite.
  - `LOGIN_ERR_UNKNOWN` - Unexpected error occurred.
  - `SESSION_TIMEOUT` - Session timed out in NetSuite.

    Note that each inbound single sign-on token includes a timestamp, and single sign-on access is only valid for 15 minutes.
- The following status code may be returned independently of the `hideloginpage` parameter value:
- `LOGOUT` - User chose to log out.

Setting Up a Single Sign-on Only Role

For security purposes, you can designate a NetSuite role as Single Sign-on Only. When a user logs in with a role that has been designated as Single Sign-on Only, validation is performed to ensure that the

user is logging in through an inbound single sign-on mechanism. This mechanism can be either the NetSuite certificate-based inbound single sign-on feature or OpenID single sign-on feature.

The Single Sign-on Only role supports strict control of credentials from the external application. This type of role increases the security of an integrated application by prohibiting a web services or UI user from accessing the system with permissions and privileges that are specifically created for inbound single sign-on access only.

Important: You cannot use NetSuite for Outlook with a Single Sign-on Only role. Users who are not sure whether their role is compatible with NetSuite for Outlook should contact their account administrator.

To designate a role as Single Sign-on Only:

1. Go to Setup > Users/Roles > Manage Roles.
2. On the Manage Roles list page, select **Edit** or **Customize** next to the role you want to set as **Single Sign-on Only**.
3. Check the **Single Sign-on Only Role** box.
4. Click **Save**.

Next, assign this role to single sign-on users as needed.

For details about setting up roles in NetSuite, see the help topic Customizing or Creating NetSuite Roles.

Mapping Users and Roles for Inbound Single Sign-on Access to NetSuite

NetSuite verifies a user's identity by comparing the remote system credentials passed in the token (company ID and user ID) to their NetSuite credentials (email, password, account, and role used to log in to NetSuite). To allow for this comparison and verification, the remote system credentials must be associated with, or mapped to, the NetSuite credentials. This mapping stores a permanent association between the user's external application identity and their NetSuite identity.

The initial single sign-on account mapping must be to an Administrator role in NetSuite. This administrator mapping serves as authorization that NetSuite trusts the external authentication system. This requirement gives NetSuite administrators control over when single sign-on functionality is available to their users. See Creating the Initial Mapping of the Administrator Role for Inbound Single Sign-on for more information.

After the initial mapping to the Administrator role is completed:

- For web store access, the administrator is required to use the web services mapSso operation to create the account mappings for multiple users so that they are available before users initiate single sign-on access.
- For NetSuite access, the administrator can use the web services mapSso operation to create the account mappings for multiple users, or can instruct users to create their own mappings.

If the administrator does not create mappings for users' external credentials and NetSuite credentials, users are required to create these mappings when they first log in with a role that requires single sign-on access. (This method of mapping is not supported for web store access.) See Creating Your Mapping for Inbound Single Sign-on to the NetSuite UI

Be aware of the following:

- If a NetSuite role used for inbound single sign-on access is deleted, the single sign-on mapping for any user with that role is automatically remapped to another role.
- If a user has a single sign-on mapping set up with a particular role and that role is removed from the user, the mapping is deleted. You can set up a new mapping for that user with a different role.
- Single sign-on mappings are not copied from a production account to a sandbox account when the sandbox is refreshed, or from one sandbox account to another. These mappings must be recreated in each sandbox account for any users who require inbound single sign-on access to that account.

> **Note:** If a user requires single sign-on access for multiple accounts, you must use a different partner account/ remote company ID for each single sign-on mapping for that user. For more information, see NetSuite Application Access Only Parameters.

Creating Your Mapping for Inbound Single Sign-on to the NetSuite UI

If your NetSuite administrator did not already create the mapping for you, the first time you log in from an external application to the NetSuite UI in your inbound single sign-on role, you must create a mapping.

> **Note:** This procedure is only for mapping access to the NetSuite application (the UI). The mapping for access to a web store must be completed by an account administrator using the mapSso operation.

To create your mapping for inbound single sign-on to the NetSuite UI:

1. Log in to the external application to be used for inbound single-sign-on access to NetSuite.
2. Click the link to go to the NetSuite UI. The NetSuite Partner Login page displays.
3. On the NetSuite Partner Login page, enter your NetSuite email address and password.
4. Click **Log in**. The NetSuite Choose a role to create the mapping page displays.
5. Click the name of the role you will use for inbound single sign-on access to this account.

   > **Note:** If you have similar roles in more than one account, ensure that you are selecting the correct role for this specific account.

   The mapping is now complete. You will automatically be logged in when accessing NetSuite by inbound single sign-on from your external application.

Technical Summary of Inbound Single Sign-on

In the following text, system A refers to an external application, and system B refers to NetSuite.

> **Note:** HTTP POST requests are not supported.

This example discusses two systems, systems A and B, and a user that has identifiers ID $_A$ and ID $_B$ in the respective systems. In the absence of single sign-on, A and B would each require ID and PASSWORD presentation for user access. In order for system A to validate the user and then use single sign-on to redirect the user to system B, without requiring further user authentication, the following steps occur:

1. System A validates user ID $_A$ as usual, requesting PASSWORD $_A$.

2. User works in system A and eventually clicks a link to system B.
3. System A creates a string T = ID_A + " " + TimeStampString.
   - ID_A is <companyID><space><userID>, so the entire token prior to encrypting and hex-encoding is <companyID><space><userID><space><timestamp>. The <companyID> in system A maps to a similar ID in system B.
   - Because a <space> is used to delimit subtokens in the token, none of the subtokens may contain <space> characters.
   - The timestamp string is a decimal representation of the number of milliseconds since January 1, 1970, 00:00:00 GMT.
4. System A encrypts T using RSA encryption with a private key, KA_{Pr}, creating a token $\{T\}KA_{Pr}$.
5. System A hex-encodes the encrypted bits so they can be transported as a URL parameter, the result being hex($\{T\}KA_{Pr}$).
6. System A directs the user's browser to a landing link on system B, including hex($\{T\}KA_{Pr}$) as a URL parameter.
7. System B hex decodes hex($\{T\}KA_{Pr}$), yielding $\{T\}KA_{Pr}$.
8. System B uses the public key, KA_{Pu}, corresponding to KA_{Pr} to retrieve T from$\{T\}KA_{Pr}$.
9. System B checks that T was recently generated by observing TimeStampString. This check reduces the risk of the token being used outside the context of single sign-on between A and B.
10. System B looks up ID_B in a table that maps $\{A,ID_A\}$ ID_B.
11. System B trusts system A's authentication procedure, and therefore logs user ID_B into system B transparently.