



PureConnect®

2022 R2

Generated:

12-July-2022

Content last updated:

27-February-2020

See [Change Log](#) for summary of changes.



Single Sign-On Identity Providers

Technical Reference

Abstract

This document describes Single Sign-On, how it is implemented with Customer Interaction Center, procedures for configuring Single Sign-On with CIC, and information on third-party identity provider services.

For the latest version of this document, see the PureConnect Documentation Library at: <http://help.genesys.com/pureconnect>.

For copyright and trademark information, see https://help.genesys.com/pureconnect/desktop/copyright_and_trademark_information.htm.

Table of Contents

Table of Contents	2
Introduction to Single Sign-On for CIC	3
Session Assertion Markup Language	4
SAML bindings and profiles	5
Supported SAML bindings and profiles	5
Web Browser Single Sign-On profile and bindings	6
Enhanced Client or Proxy (ECP) profile	7
CIC implementation of SAML	8
Protocols	8
Service provider subsystems	8
Security token creation	9
Single Sign-On configurations	9
Single CIC server	9
CIC Switchover pair	10
Dedicated CIC STS	10
Certificates	11
HTTPS digital certificate	11
Secure Token Server validation digital certificate	13
Prerequisite tasks for Single Sign-On in CIC	14
Select identity provider method	14
Microsoft AD FS identity provider	14
Third-party identity provider services	14
Customer Interaction Center user database	14
Determine issuer/provider name/relying party identifier/partner identifier/entity ID	15
Determine address scheme for certificates and tokens	15
Determine token expiration period	15
Determine default port for Single Sign-On	15
Gather CIC server endpoint information	16
Assertion Consumer Service URL	16
ACS URL example configurations	18
Initiate selected identity provider method	18
Gather identity provider information	18
Configure Single Sign-On for a CIC system	20
Enable Single Sign-On authentication on the CIC server	20
Configure the CIC server as the service provider	23
Configure Secure Token Server	23
Administer HTTPS certificate for the CIC service provider	26
Configure identity provider settings for CIC	39
Copy identity provider validation certificates to the CIC server	40
Ensure the format of the validation certificates	40
Configure identity provider settings in Interaction Administrator	45
Provide CIC Single Sign-On information to identity provider	63
Single Sign-On Configuration Utility	63
Enable SSO Configuration Utility through Interaction Administrator	63
Use SSO Configuration Utility on a client workstation	65
Configure Microsoft AD FS as an identity provider	70
Configure PingOne as an identity provider	78
Configure Salesforce as an identity provider	78
Test Single Sign-On for the identity provider	78
Configure CIC client application workstations	81
Import the HTTPS certificate of the CIC server onto workstations hosting CIC client applications	82
Use group policies to import the CIC HTTPS certificate onto workstations	82
Manual import of the CIC HTTPS certificate onto workstations	82
Test the imported HTTPS certificate	82
Troubleshooting	83
Single Sign-On troubleshooting tools	83
Examine log files	83
Updating the CIC server causes certificate validation issues	83
SAML 2.0 message exchange example	85
Change Log	89

Introduction to Single Sign-On for CIC

Single Sign-On is an industry term for using one instance of user identity authentication across multiple applications and systems. Customer Interaction Center has developed Single Sign-On capabilities for many of its client applications and multiple third-party validation services.

Note: Some Customer Interaction Center subsystems and add-ons maintain user credentials separate from those of the CIC server and do not recognize Single Sign-On security tokens. Examples of CIC subsystems and add-ons that do not support the Single Sign-On feature of CIC are Interaction SIP Proxy, Interaction Media Server, Interaction Edge, and Interaction Web Portal.

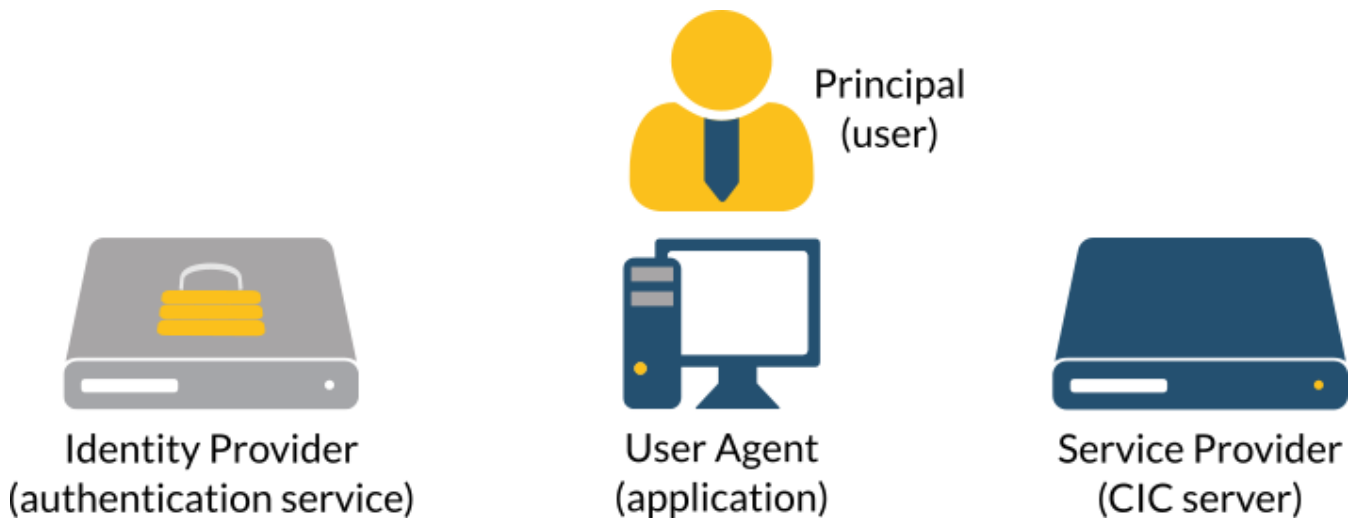
Session Assertion Markup Language

For Single Sign-On, Customer Interaction Center follows the SAML (Security Assertion Markup Language) version 2 standard, maintained by Organization for the Advancement of Structured Information Standards (OASIS).

SAML is an open, standard protocol, based on XML, for exchanging authentication data. SAML defines only the structure, elements, and assertions in messages, including security tokens. SAML does not define how user credentials are authenticated, which is delegated to the applications, systems, and services involved.

The SAML standard defines the following roles:

User Agent/Principal	This component is the application through which a person or entity—called the <i>Principal</i> —provides authentication credentials for accessing a system. Examples of user agents for CIC are Interaction Desktop or IC Business Manager.
Service Provider	The system that provides application services to the user, such as the CIC server.
Identity Provider	The application or entity that validates or rejects the authentication of the user credentials for the service provider. Based on the result from the identity provider, the service provider either allows or denies access to the user agent.



In SAML, all communications are conducted through the user agent. The service provider and identity provider do not communicate directly. This method provides an added layer of security for service providers.

SAML bindings and profiles

To exchange messages through a communication protocol, SAML uses *bindings*. For example, sending SAML messages to an entity using the Simple Object Access Protocol (SOAP). These bindings describes how SAML messages can be mapped to the message format of the communication protocol.

While exchanging messages with a communication protocol, it may be necessary to embed or extract SAML assertions. A set of rules that defines how those assertion actions are conducted for a communication protocol are *profiles*.

- [Supported SAML bindings and profiles](#)
- [Web Browser Single Sign-On profile and bindings](#)
- [Enhanced Client or Proxy \(ECP\) profile](#)

Supported SAML bindings and profiles

CIC supports the following SAML 2.0 binding and profile implementations:

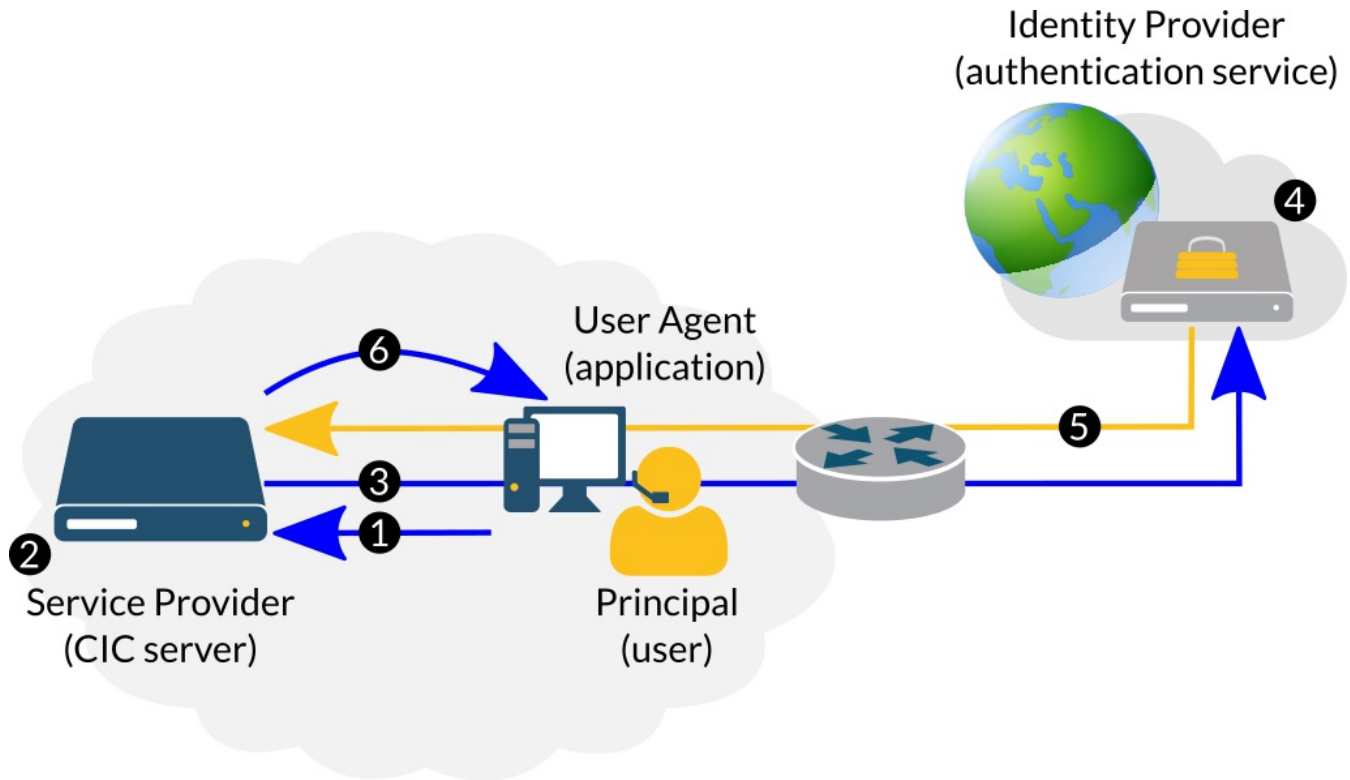
- Web Browser Single Sign-On profile with the HTTP POST binding
- Web Browser Single Sign-On profile with the HTTP Redirect binding
- Enhanced Client or Proxy (ECP) profile

Note: Starting with CIC 2016 R1, client applications that are based on IceLib, such as Interaction Desktop, IC Business Manager, and IC Server Manager now supports custom HTML forms-based authentication as well as Command Access Card (CAC) or Smart Card authentication. These forms can be the same as those displayed with web-based CIC client applications.

Web Browser Single Sign-On profile and bindings

The Web Browser Single Sign-On profile is common in web applications and includes the following bindings:

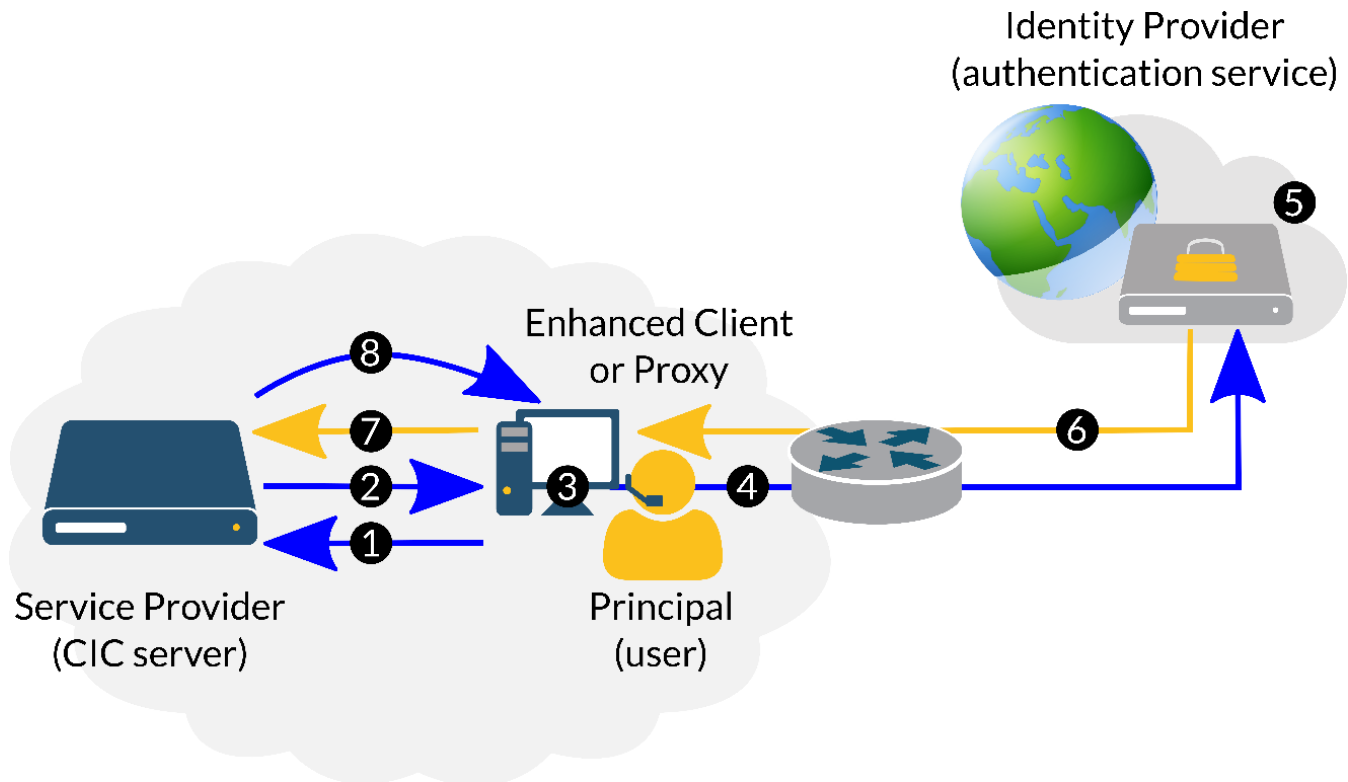
- HTTP Post
- HTTP Redirect



Step	Description
1	The User Agent attempts to access a resource on the Service Provider.
2	The Service Provider determines which Identity Provider should authenticate the access request.
3	The Service Providers sends an <AuthnRequest> message to the selected Identity Provider through the User Agent.
4	The Identity Provider authenticates the Principal (user credentials).
5	The Identity Provider issues a <Response> message to the Service Provider through the User Agent.
6	The Service Provider either allows or denies the access request to the User Agent based on the <Response> message from the Identity Provider.

Enhanced Client or Proxy (ECP) profile

Another profile is the Enhanced Client or Proxy (ECP) profile. Non-web-based CIC applications, such as CIC clients based on the Microsoft .NET Framework, use the ECP Profile.



Step	Description
1	The Enhanced Client or Proxy (ECP) attempts to access a resource on the Service Provider, using Principal credentials, through HTTPS protocol.
2	The Service Provider sends an <AuthnRequest> message to the ECP.
3	The ECP determines which Identity Provider to use for authenticating the Principal credentials.
4	The ECP sends the <AuthnRequest> message to the selected Identity Provider using the SAML SOAP binding.
5	The Identity Provider authenticates the Principal (user credentials).
6	The Identity Provider issues a <Response> message to the ECP.
7	The ECP sends the <Response> message from the Identity Provider to the Service Provider.
8	The Service Provider either allows or denies the access request to the ECP based on the <Response> message from the Identity Provider.

CIC implementation of SAML

Protocols

Customer Interaction Center can use two protocols for relaying SAML-based, Single Sign-On messages between itself, the service provider, and an application (user agent):

- Notifier - A Genesys secure communications protocol for CIC, its subsystems, and applications, the Notifier protocol is used by Windows-based CIC applications and subsystems. These CIC applications and subsystems use the Microsoft .NET Framework, including some .NET Framework components related to Single Sign-On. The Icelib API also uses the .NET Framework.
- HTTPS - A standard, secure protocol that is commonly used by web-based applications, including some web-based CIC client applications.

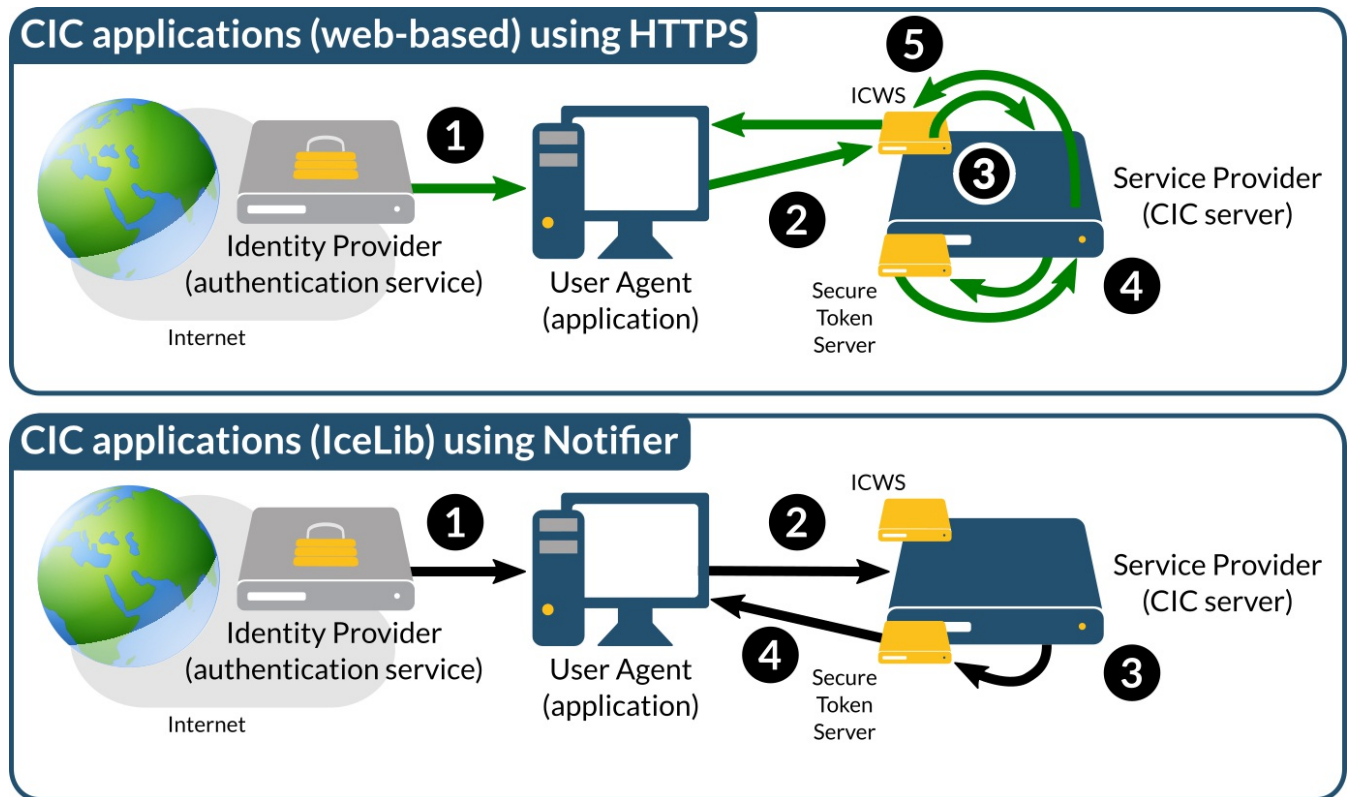
Service provider subsystems

As a service provider, Customer Interaction Center contains the following subsystems to support Single Sign-On:

Subsystem	Description
Secure Token Server	After the CIC server (the service provider) receives a successful validation of user credentials, it provides a token to the user agent that it can store for subsequent access requests. The Secure Token Server subsystem creates these tokens and securely transmits them to the user agent. Tokens are temporary as they can expire after a period of elapsed time, and are discarded when the user logs off or the PC/device has been restarted.
Interaction Center Web Services (ICWS)	For web browser-based applications that use the HTTPS protocol for secure communications, ICWS is a subsystem that acts as a service provider proxy. It converts messages between the HTTPS protocol and the Notifier protocol, and relays them to and from the CIC server.

Security token creation

The following diagrams display how CIC generates a secure token for CIC applications based on validated user credentials.



For web-based CIC applications, IC Web Service (ICWS) acts as a service provider proxy, which then interacts with the CIC server and the Secure Token Server subsystem to return a secure token to web-based CIC applications.

For CIC applications that are based on the Microsoft .NET Framework and the IceLib PureConnect library, the proprietary Notifier protocol facilitates communication with the CIC server (service provider), its Secure Token Server subsystem, and the CIC client application for acquiring a secure token.

If a user successfully completes the SSO credential authentication for a CIC client application (user agent), starting a different CIC client application of the same type on the same system would pick up the existing security token and send it to the Secure Token Server so that the application can have access to CIC resources.

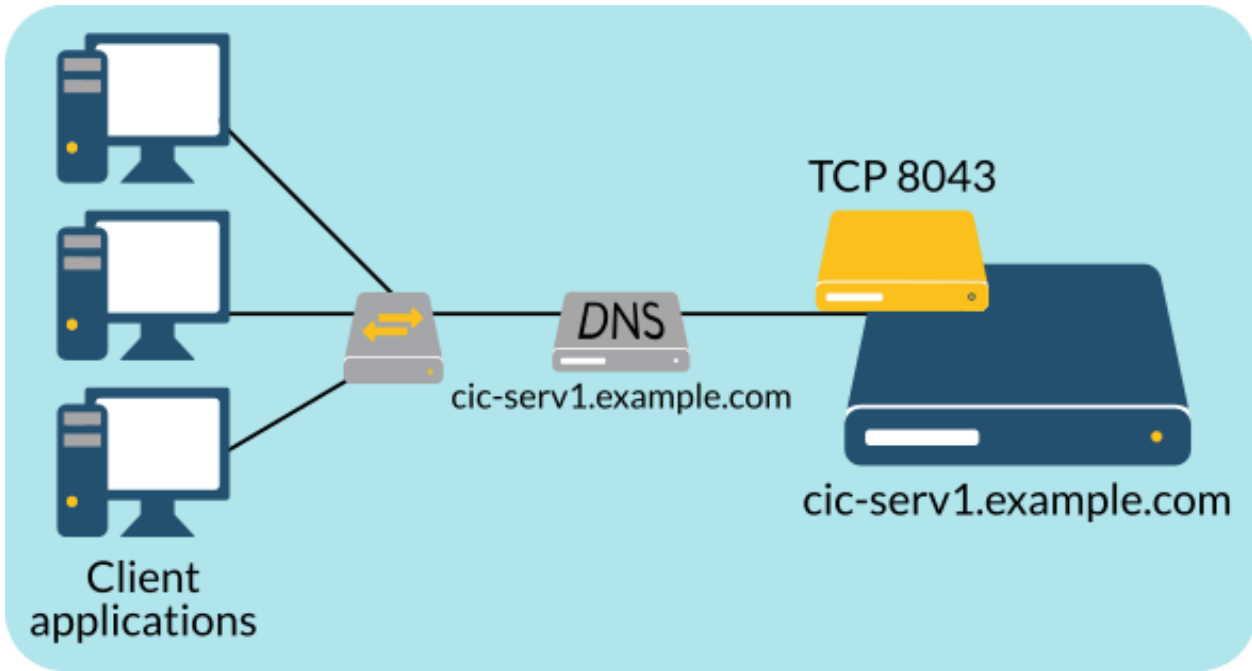
Single Sign-On configurations

Depending on your CIC deployment, network environment, and how you want Single Sign-On to work within it, you can use different configurations. Each of these configurations can result in a different *machine name* for the Security Token Server subsystem of the CIC server.

The *machine name* is an address through which user agents (Single Sign-On client applications) can contact the Secure Token Server subsystem of your CIC deployment for the purpose of communicating SAML messages.

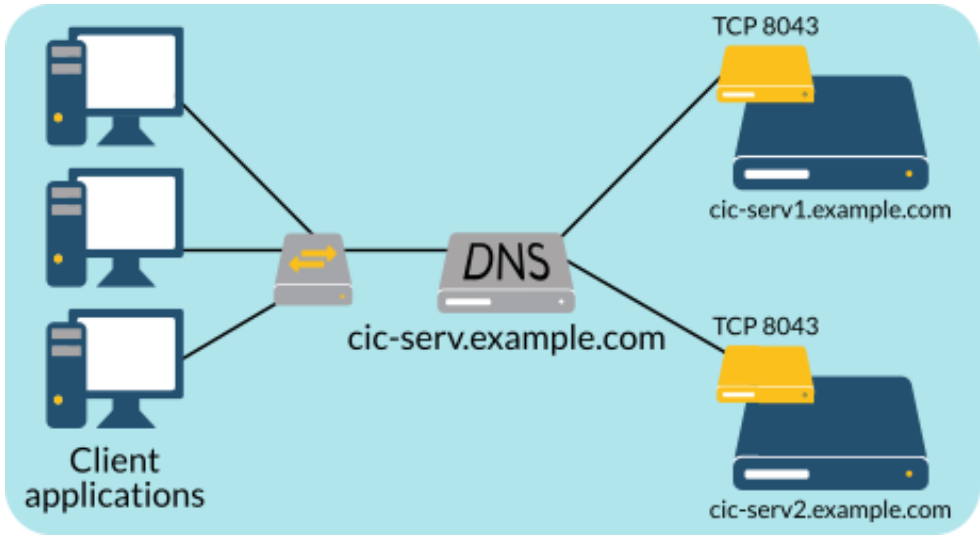
Single CIC server

In this configuration, the FQDN of the CIC server (cic-serv1.example.com) is the machine name.



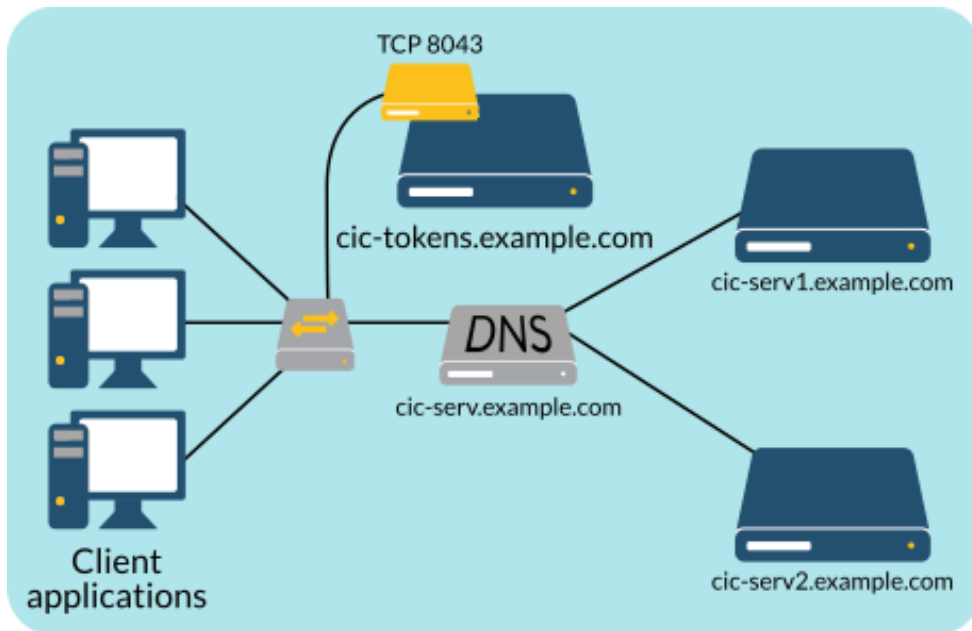
CIC Switchover pair

In this configuration, the DNS SRV, DNS A, or DNS AAAA record for the CIC switchover pair (cic-serv.example.com) is the machine name.



Dedicated CIC STS

In this configuration, the FQDN of a separate CIC server for only servicing tokens (cic-tokens.example.com) is the machine name.



For this configuration to function correctly, the certificates and private keys in the following directories must be the same on all three CIC servers:

- `\I3\IC\Certificates\HTTPS`
- `\I3\IC\Certificates\ICSecureTokenServer`

Certificates

An important aspect of any Single Sign-On implementation is that of digital certificates, which ensure trusted communications from known network entities, including CIC servers. The CIC Single Sign-On solution uses the following digital certificates:

- [HTTPS digital certificate](#)
- [Secure Token Server validation digital certificate](#)

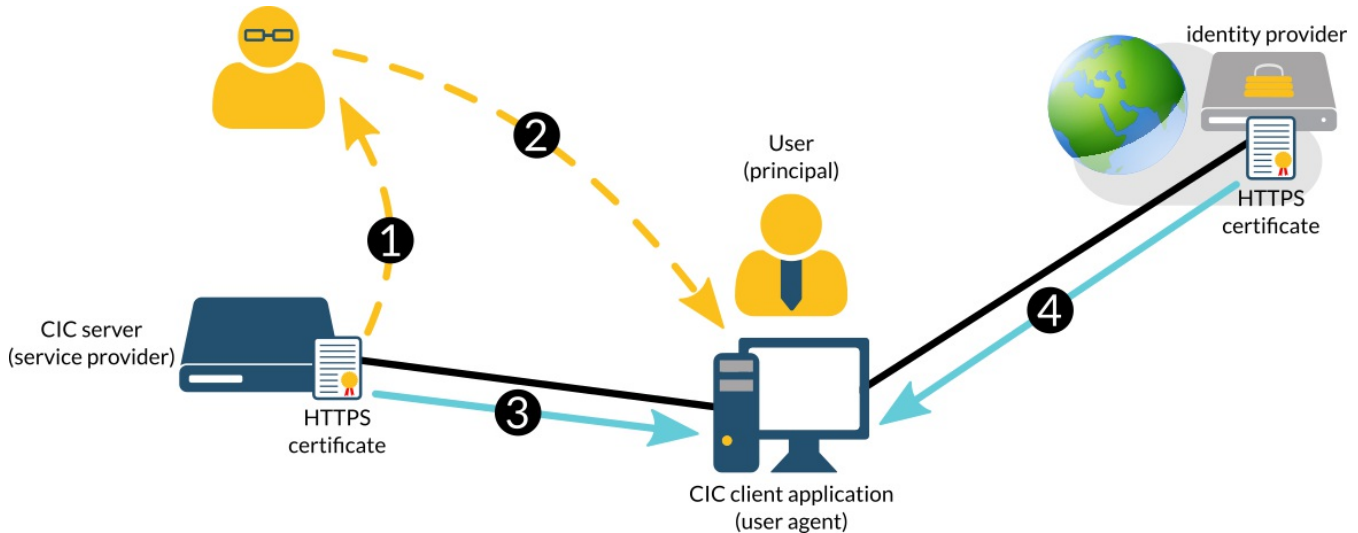
HTTPS digital certificate

This certificate provides trusted, secure communications between the user agent (CIC client application) and the service provider (CIC server).

Each time that the CIC server starts, it searches for its HTTPS certificate. If it does not find the HTTPS certificate, it creates it, a private encryption key, and a public encryption key, using the Fully Qualified Domain Name (FQDN).

Tip: Genesys recommends using FQDN addressing for all servers and subsystems entities in a CIC environment.

The following diagram shows how a CIC Single Sign-On environment uses HTTPS certificates:



Step	Description
1	Before a CIC client application can trust the messages from a CIC server, the administrator must copy the HTTPS certificate from the CIC server.
2	The administrator imports the HTTPS certificate from the CIC server into the Trusted Root Certificate Authorities Certificate Store on each machine that will host a CIC client application for Single Sign-On. A common method of importing the HTTPS certificate of the CIC server to client workstation is that of Group Policies through Microsoft Active Directory.
3	The CIC client application, now updated with the HTTPS certificate of the CIC server, can validate and trust communications from the CIC server.
4	The identity provider sends its own HTTPS certificates in Single Sign-On communications with the CIC client. Since the Trusted Root Certificate Authorities Certificate Store of the machine hosting the CIC client application has already has entries for most Certificate Authorities, the CIC client application can validate and trust communications from the identity provider.

Important!

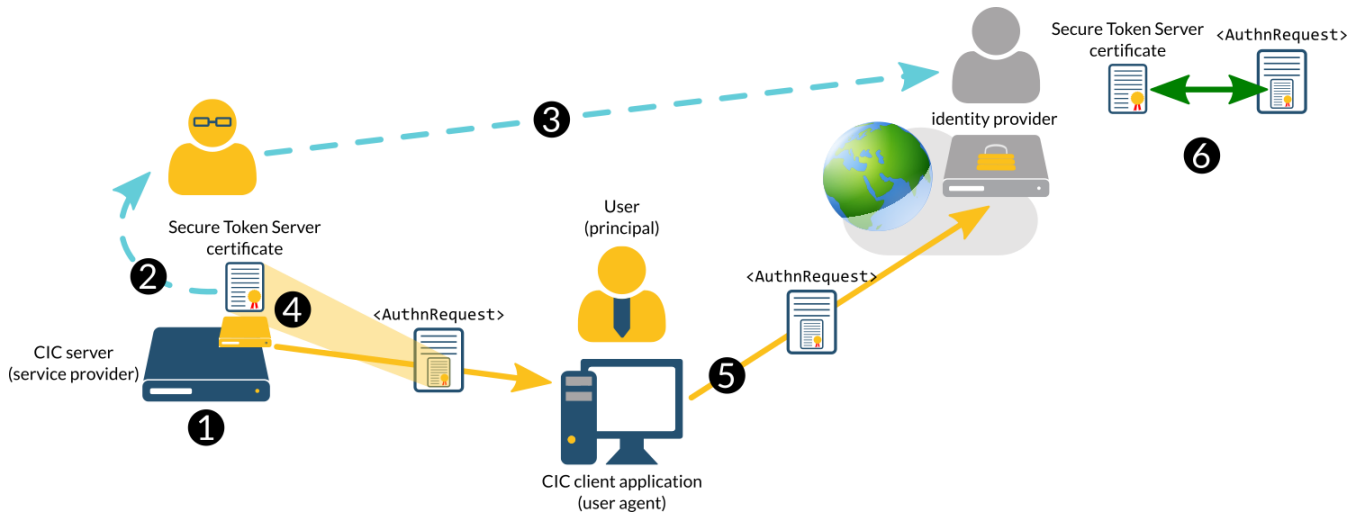
The address that the CIC client application uses to access CIC server resources must match the address within the HTTPS certificate in its Trusted Root Certificate Authorities Certificate Store of the workstation hosting the CIC client application.

For example, if the CIC client application attempts to use a resource on the CIC server through a DNS A record of `cic-server.example.com` and the certificate in the Trusted Root Certificate Authorities Certificate Store of the workstation was generated with `cic-serv1.example.com` for a specific CIC server in a switchover pair, the validation of the certificate fails.

Secure Token Server validation digital certificate

This certificate, if *signed authentication requests* are required by the identity provider, is embedded within SAML <AuthnRequest> messages. Each identity provider may have differing configurations and requirements. As such, you may need to provide the Secure Token Server validation certificate to the identity provider, even if it does not require signed authentication requests.

The following diagram shows how a CIC Single Sign-On environment uses Secure Token Server validation certificates:



Step	Description
1	The CIC server, as a Single Sign-On service provider, generates a certificate for the Secure Token Server subsystem with which it can sign SAML <AuthnRequest> messages.
2	The administrator copies the Secure Token Server certificate from the CIC server.
3	The administrator sends the Secure Token Server certificate to the identity provider.
4	For signed SAML <AuthnRequest> messages, the Secure Token Server subsystem embeds the certificate in the message and sends it to the user agent when it requests authorization.
5	The user agent sends the SAML <AuthnRequest> message, which contains the certificate, to the identity provider.
6	The identity provider compares the certificate in the <AuthnRequest> message with the one that was previously provided by the administrator to determine if it can be trusted. If the certificate can be trusted, the identity provider then processes the SAML <AuthnRequest> message and attempts to validate the principal.

Prerequisite tasks for Single Sign-On in CIC

Select identity provider method

For your CIC environment, you can use one of the following identity providers that support the Session Assertion Markup Language (SAML) authentication standard:

- Microsoft Active Directory Federation Services (AD FS)
- An Internet-based, third-party identity provider service
- Customer Interaction Center user database

The type of identity provider that you select determines what information about your service provider that you must collect and provide for initial configuration.

Important!

For PureConnect Cloud customers, the Uniform Resource Locator (URL) address for your identity provider must be accessible by your client workstations that host the CIC applications that will use Single Sign-On.

Microsoft AD FS identity provider

Microsoft Active Directory Federation Services (AD FS) enables you to create your own Single Sign-On identity provider, which you can install on-premises within an Active Directory domain.

Both Microsoft AD FS 2.0 and 3.0 support Windows Authentication over HTTPS and web browser-based authentication. However, CIC client applications support only Microsoft AD FS 3.0 for web browser-based authentication. CIC client applications support both Microsoft AD FS 2.0 and 3.0 for Windows Authentication over HTTPS.

Third-party identity provider services

- PingOne
- IBM LightHouse
- Salesforce.com
- Other SAML 2.0-compliant identity provider services

Note: Genesys previously verified CIC Single Sign-On compatibility with the open-source identity provider Shibboleth.net (<https://shibboleth.net>). However, Genesys does not provide technical support for configuring CIC Single Sign-On with Shibboleth.

Customer Interaction Center user database

If your Single Sign-On goals involve only CIC clients, you can use the CIC server as both a service provider and an identity provider. CIC clients can use the same security token that the CIC server provides after it validates the credentials of the principal (user) through its database. You do not need to collect any service provider information to provide to the identity provider, which is the same system for this method.

Note: Some CIC subsystems, such as Interaction Media Server and Interaction SIP Proxy, contain their own user databases for accessing the web-based administration interface. These CIC subsystems are not included in the CIC Single Sign-On solution as most CIC users do not require access to these subsystems.

Determine issuer/provider name/relying party identifier/partner identifier/entity ID

In a Single Sign-On environment, a CIC client application (user agent) must be able to do the following actions:

- Trust the HTTPS communications from the CIC server

You use the default, generated HTTPS certificate or generate the HTTPS certificate on the CIC server (service provider), and import it into the client host computer.

- Communicate with the Secure Token Server subsystem of the CIC server (service provider)

To facilitate these actions, the identity of the service provider must be established. In the CIC Single Sign-On environment, this item is the address and port number of the CIC server (service provider) and is commonly called by different names:

- Issuer
- Provider name
- Relying party identifier
- Partner identifier
- Entity ID

Determine address scheme for certificates and tokens

Your network addressing scheme must be robust so that no irregularities will change future accessibility for these actions. System restarts that change IP addresses or CIC server switchovers without appropriate DNS records are some examples of configurations that interrupt not only Single Sign-On communications but basic CIC communications, including Voice-over-IP.

Important!

Because of possible variations in IP addressing through Dynamic Host Control Protocol (DHCP), Domain Name Service (DNS) entries, Network Address Translation (NAT), IPv4/IPv6 differences, IPv6 address truncation, unreachable short host names, and other considerations, Genesys recommends that you use only Fully Qualified Domain Name (FQDN) addresses in your CIC environment.

The FQDN address scheme also simplifies creation and ensures consistency for the machine name and the [Assertion Consumer Service URL](#) address.

Determine token expiration period

When a CIC client acquires a validation token, it retains that validation token for a set period of time that you configure. By default, the period of time is 14 days. You can modify this value to align with the Single Sign-On requirements for your organization.

You configure the token expiration period in the CIC Secure Token Server through Interaction Administrator. For more information on configuring the CIC Secure Token Server, see [Configure Secure Token Server](#).

Determine default port for Single Sign-On

You must also determine if the default port of 8043 on the CIC server (service provider) will not conflict with any other usage of that network port or within your network, such as firewall settings, Network Address Translation, Session Border Controllers, and so on. If you plan to use ICWS in your CIC environment, you must ensure that the default port of 8019 will not conflict with the same network entities. If you later decide to change the port number, you must create a new ACS URL address and supply it to the identity provider when you change the port number on the CIC or ICWS server and any involved network entities.

Gather CIC server endpoint information

For Single Sign-On HTTPS certificates and security tokens, you must gather endpoint information about the CIC server. Your selected identity provider needs the following service provider endpoint information for initial configuration of its service for your CIC SSO environment:

- Machine name - The address of the CIC server (service provider) whose Secure Token Server subsystem will service SAML messages for Single Sign-On.

For more information about possible CIC configurations that use different addresses for the *machine name*, see [Single Sign-On configurations](#).

- [Assertion Consumer Service URL](#)

Assertion Consumer Service URL

One required piece of information that you must provide to the identity provider is the *Assertion Consumer Service (ACS)* URL address, which the identity provider will use to verify that the SAML messages from that service provider can be serviced. Otherwise, the identity provider will ignore it as a DDoS attack. The ACS URL is a combination of the Secure Token Server subsystem address, its port number for handling SAML messages, the SAML binding, and any necessary information that is specific for CIC or ICWS.

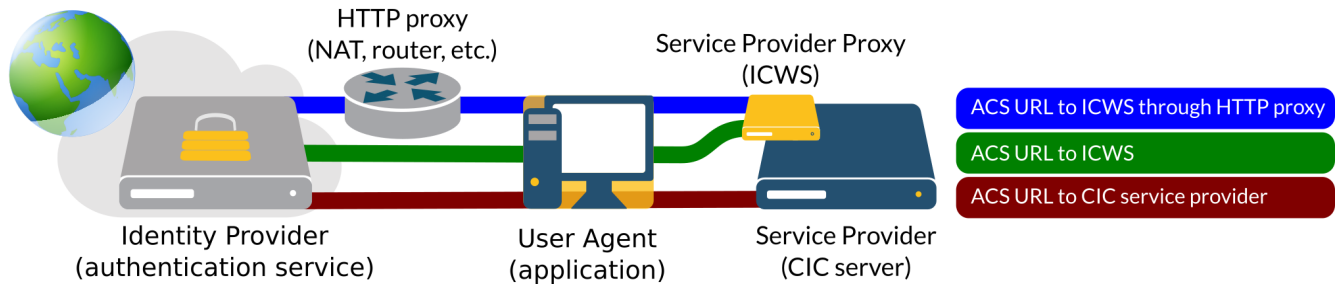
ACS URL address use the following syntax:

`https://SecureTokenServer.Port/AuthenticationType`

The following table describes each portion of the ACS URL for your service provider:

Item	Service Provider	Description
SecureTokenServer	CIC	The address of the CIC server that hosts the Secure Token Server subsystem through which CIC will issue security tokens
	ICWS	<p>The address of the ICWS server, which will function as the Single Sign-On service provider</p> <div style="border: 1px solid orange; padding: 5px;"> <p>Important!</p> <p>As it is possible to use an HTTP proxy with ICWS and Interaction Connect, the ACS URL will vary. Genesys recommends the following syntax for this configuration:</p> <ul style="list-style-type: none"> • <code>baseURL/api/server</code> • <code>baseURL</code> represents the address of the application. • <code>server</code> represents the address of the ICWS server. </div> <p>Examples:</p> <ul style="list-style-type: none"> • Through HTTP proxy: icws.example.com • Without HTTP proxy: connect.example.com/api/icws.example.com <div style="border: 1px solid gray; padding: 5px;"> <p>Note: If you have multiple ICWS servers, you should provide an ACS URL for each one, if allowed by the identity provider.</p> </div>
Port	CIC	<p>The network port through which the Secure Token Server subsystem of the CIC server will listen for SAML messages</p> <p>The default value is 8043.</p>
	ICWS	<p>The network port through which Interaction Center Web Services (ICWS) will listen for SAML messages</p> <p>The default value is 8019.</p> <div style="border: 1px solid gray; padding: 5px;"> <p>Note: If you are using an HTTP proxy in your network, you do not need to specify a port number for the ACS URL address.</p> </div>
AuthenticationType	CIC	<p>The SAML protocol and binding for validating the credentials of the principal</p> <p>Examples:</p> <ul style="list-style-type: none"> • SAML2WebBrowserPostHTTPS/login • SAML2WebBrowserRedirectHTTPS/login
	ICWS	/icws/connection/single-sign-on/return

ACS URL example configurations



ICWS with HTTP proxy ACS URL example

`https://connect.example.com/api/icws.example.com/icws/connection/single-sign-on/return`

ICWS ACS URL example

`https://icws.example.com:8019/icws/connection/single-sign-on/return`

CIC ACS URL example

`https://cic.example.com:8043/SAML2WebBrowserPostHTTPS/login`

Initiate selected identity provider method

Depending on which identity provider method you select, you must do one of the following tasks before you can configure your Customer Interaction Center environment for Single Sign-On:

- Install Microsoft Active Directory Federation Service (AD FS) within your network.
- Acquire the services of the third-party identity provider and submit the service provider endpoint information that you accumulated in [Gather CIC server endpoint information](#).

Important!

Ensure that any third-party identity provider service supports one or more of the SAML bindings and profiles that are available for CIC Single Sign-On. For more information, see [SAML bindings and profiles](#).

- If you selected to use the CIC server as both the service provider and the identity provider, you may still need to configure a different addressing scheme so that workstations hosting CIC applications are able to access the Secure Token Server subsystem of the CIC server. One example is where workstations are in a different domain than the CIC server and its Secure Token Server subsystem.

Gather identity provider information

After you have selected an identity provider to use with your CIC Single Sign-On implementation, you must gather information from the identity provider. You will need the following information when you configure the CIC server as the service provider:

Item	Description												
SAML 2.0 metadata XML file	Starting with CIC 2015 R4, you can use an Interaction Administrator feature to import an XML file that contains the necessary information for SAML SSO communications with the third-party identity provider.												
List of supported SAML 2.0 profiles and binding implementations	This list can be useful if, in the future, you decide to change or add another profile and binding implementation in your CIC Single Sign-On environment.												
Identity Provider signing requirement	Does the identity provider require that <AuthnRequest> SAML messages be signed (embedded signature and X.509 certificate)?												
Additional <AuthnRequest> Identity Provider requirements	<p>Determine if the identity provider requires any of the following SAML attributes:</p> <table border="1" data-bbox="337 615 946 827"> <tbody> <tr> <td>• ID</td> <td>• AssertionCustomerServiceIndex</td> </tr> <tr> <td>• Version</td> <td>• AssertionConsumerServiceURL</td> </tr> <tr> <td>• Consent</td> <td>• AttributeConsumingServiceIndex</td> </tr> <tr> <td>• ForceAuthn</td> <td>• ProviderName</td> </tr> <tr> <td>• IsPassive</td> <td>• NameIDPolicy</td> </tr> <tr> <td>• ProtocolBinding</td> <td></td> </tr> </tbody> </table> <p>If your identity provider requires SAML attributes, enter them through the SAML Attributes tab of the Configuration dialog box for a SAML profile and binding for the identity provider. Step 11 of the Manually configure identity provider settings procedure addresses this aspect.</p>	• ID	• AssertionCustomerServiceIndex	• Version	• AssertionConsumerServiceURL	• Consent	• AttributeConsumingServiceIndex	• ForceAuthn	• ProviderName	• IsPassive	• NameIDPolicy	• ProtocolBinding	
• ID	• AssertionCustomerServiceIndex												
• Version	• AssertionConsumerServiceURL												
• Consent	• AttributeConsumingServiceIndex												
• ForceAuthn	• ProviderName												
• IsPassive	• NameIDPolicy												
• ProtocolBinding													
Identity Provider URL address	<p>Depending on which identity provider method you selected, acquire the URL address to which the CIC client application (user agent) will send all SAML response messages:</p> <ul style="list-style-type: none"> • For a Microsoft AD FS server that is installed on-premises, this address is the FQDN of the AD FS server as seen by workstations hosting CIC Single Sign-On applications. • For an Internet-based identity provider, this address is its Internet URL address. • If you are using the same CIC server as the identity provider, you do not need to gather any information. 												
Identity Provider validation certificate	<p>The certificate that the CIC server will use to validate all SAML response messages from the identity provider.</p> <p>Contact your identity provider or consult the documentation for your identity provider for information on how to obtain the validation certificate.</p>												
Identity Provider claims	<p><i>Claims</i> are assertion attributes that identity providers include in SAML response messages. These claims represent identifying or conditional information associate with an authentication request, such as the Windows account name of the requesting user, an e-mail address, user role, expiration time periods, computer network environment information, and many others.</p> <p>For example, the identity provider could include the following assertion attribute in their SAML response messages:</p> <pre data-bbox="337 1545 1203 1776"><AttributeStatement> <Attribute Name="http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"> <AttributeValue>EXAMPLEDOMAIN\DomainAdmin</AttributeValue> </Attribute> </AttributeStatement></pre> <p>For the CIC server acting as the Single Sign-On service provider, the CIC server must be able to equate the http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname SAML attribute with a CIC user ID with a matching CIC user attribute.</p>												

Configure Single Sign-On for a CIC system

As Customer Interaction Center uses the Security Assertion Markup Language (SAML) 2.0 standard for Single Sign-On functionality, you must configure the service provider, identity provider, and user agent components of the SAML model in the following order:

[Enable Single Sign-On authentication on the CIC server](#)

[Configure the CIC server as the service provider](#)

[Configure identity provider settings for CIC](#)

[Single Sign-On Configuration Utility](#)

[Configure Microsoft AD FS as an identity provider](#)

[Configure PingOne as an identity provider](#)

[Configure Salesforce as an identity provider](#)

[Test Single Sign-On for the identity provider](#)

[Import the HTTPS certificate of the CIC server onto workstations hosting CIC client applications](#)

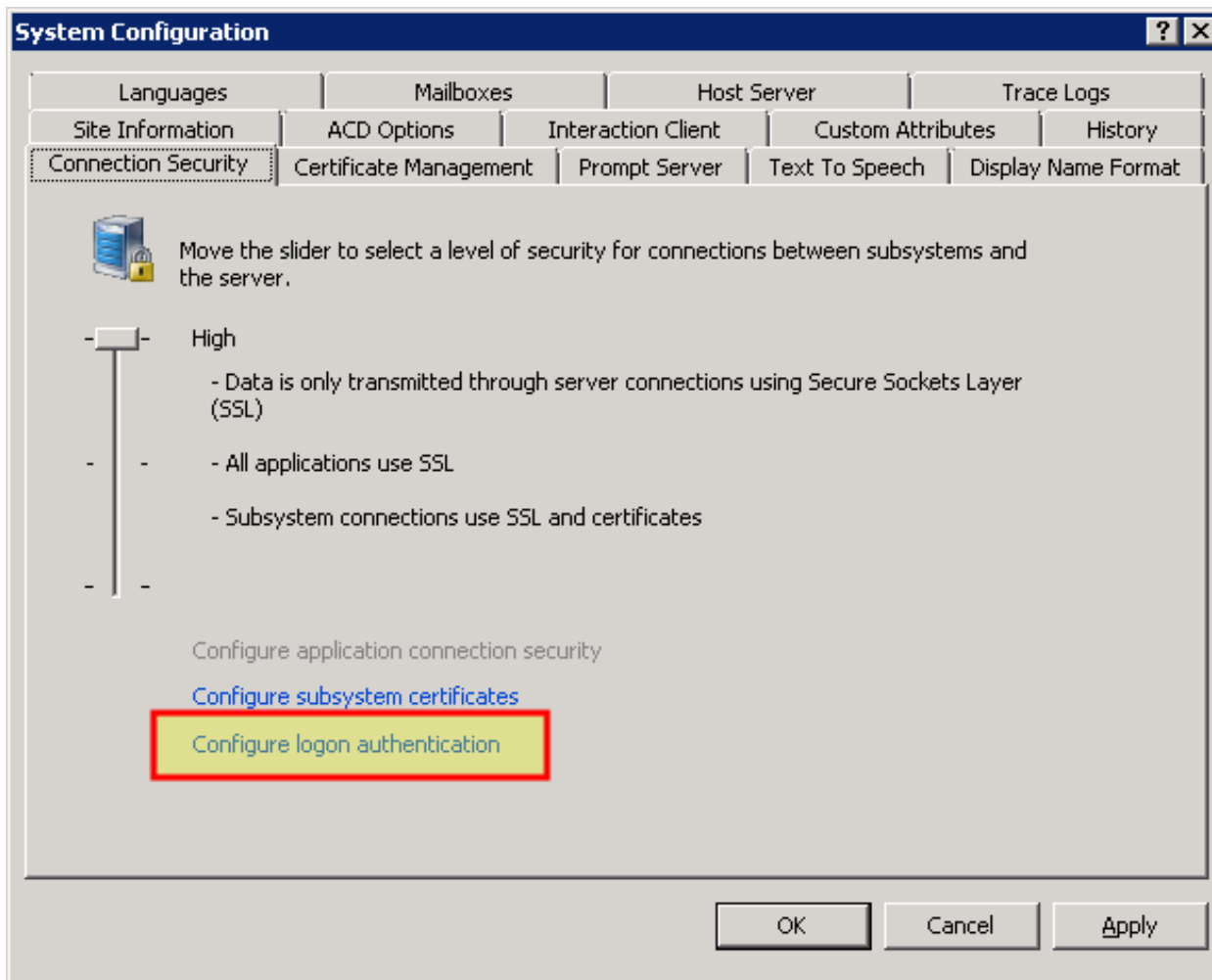
Enable Single Sign-On authentication on the CIC server

1. Open Interaction Administrator.
2. In the left pane of the **Interaction Administrator** window, select the **System Configuration** object.
3. In the right pane, double-click the **Configuration** item.

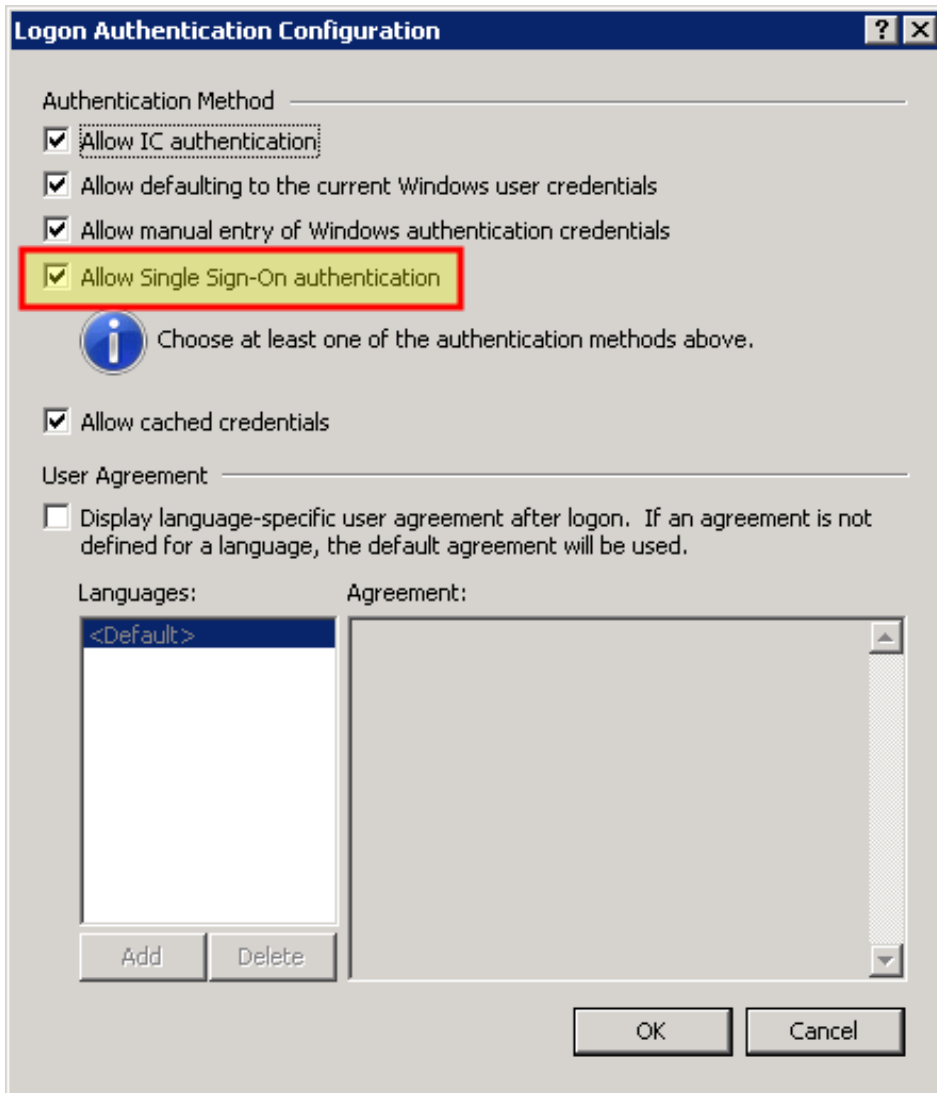
Interaction Administrator displays the **System Configuration** dialog box.

The screenshot shows the 'System Configuration' dialog box. The 'Site Information' tab is selected. The main area contains the text: 'Enter the name of your company, and an optional location name.' Below this text are two text input fields. The first field is labeled 'Organization Name:' and contains the text 'OrganizationName'. The second field is labeled 'Location Name:' and contains the text 'Location 111'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

4. On the **System Configuration** dialog box, select the **Connection Security** tab.
5. On the **Connection Security** tab, select the **Configure logon authentication** hyperlink.



Interaction Administrator displays the **Logon Authentication Configuration** dialog box.



6. In the **Authentication Method** group, ensure that the **Allow Single Sign-On authentication** check box is enabled.

Important!

You must have at least one authentication method enabled in the **Logon Authentication Configuration** dialog box so that the CIC server can authenticate users.

Note: You can use multiple authentication methods in conjunction with Single Sign-On. Genesys recommends that you thoroughly test your Single Sign-On environment after configuration before you disable any other authentication methods. Successful testing ensures that users of CIC client applications are not prevented from successfully logging on to the CIC server.

7. After you enable the necessary logon features in the **Logon Authentication Configuration** dialog box, select the **OK** button.
8. In the **System Configuration** dialog box, select the **OK** button.

The CIC server is now configured to allow Single Sign-On logon authentication.

Configure the CIC server as the service provider

To configure a Customer Interaction Center server as a SAML service provider, complete the following tasks in order:

[Configure Secure Token Server](#)

[Administer HTTPS certificate for the CIC service provider](#)

[Generate a self-signed HTTPS certificate for non-FQDN configurations](#)

[Generate a non-signing HTTPS certificate for FQDN configurations](#)

[Generate a non-signing HTTPS certificate for non-FQDN configurations](#)

[Replace the CIC HTTPS certificate with an externally-generated certificate](#)

[Copy identity provider validation certificates to the CIC server](#)

[Ensure the format of the validation certificates](#)

[Configure identity provider settings in Interaction Administrator](#)

[Import SAML 2.0 metadata from identity provider](#)

[Manually configure identity provider settings](#)

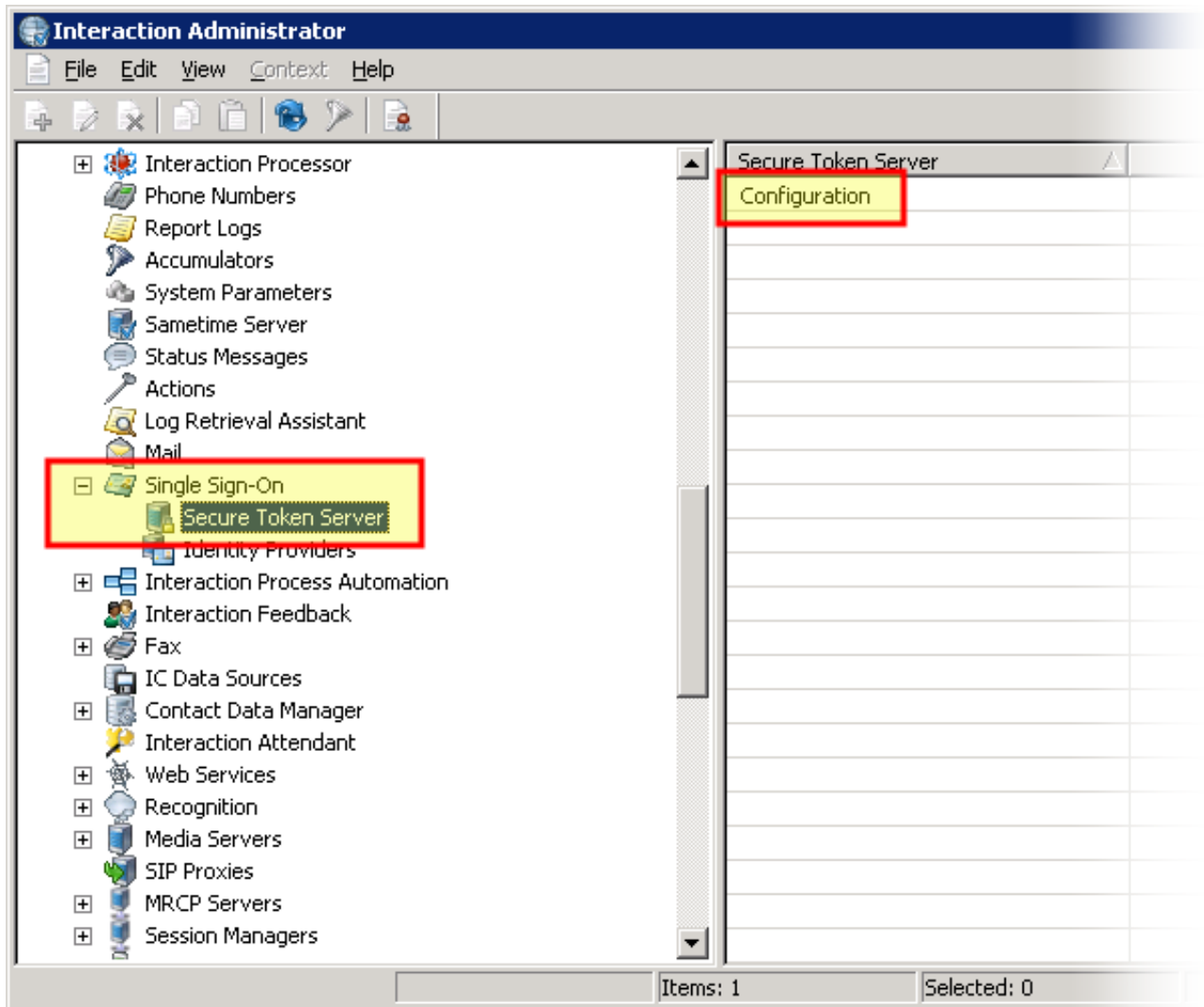
Configure Secure Token Server

The CIC server contains a Secure Token Server subsystem that issues security tokens to user agents (applications) that supply successful principal (user) authentication. User agents (CIC client applications) can then provide that security token to the service provider (CIC server) for subsequent requests to access its other resources or subsystems.

1. Ensure that you have the following information for the service provider endpoint (CIC server):

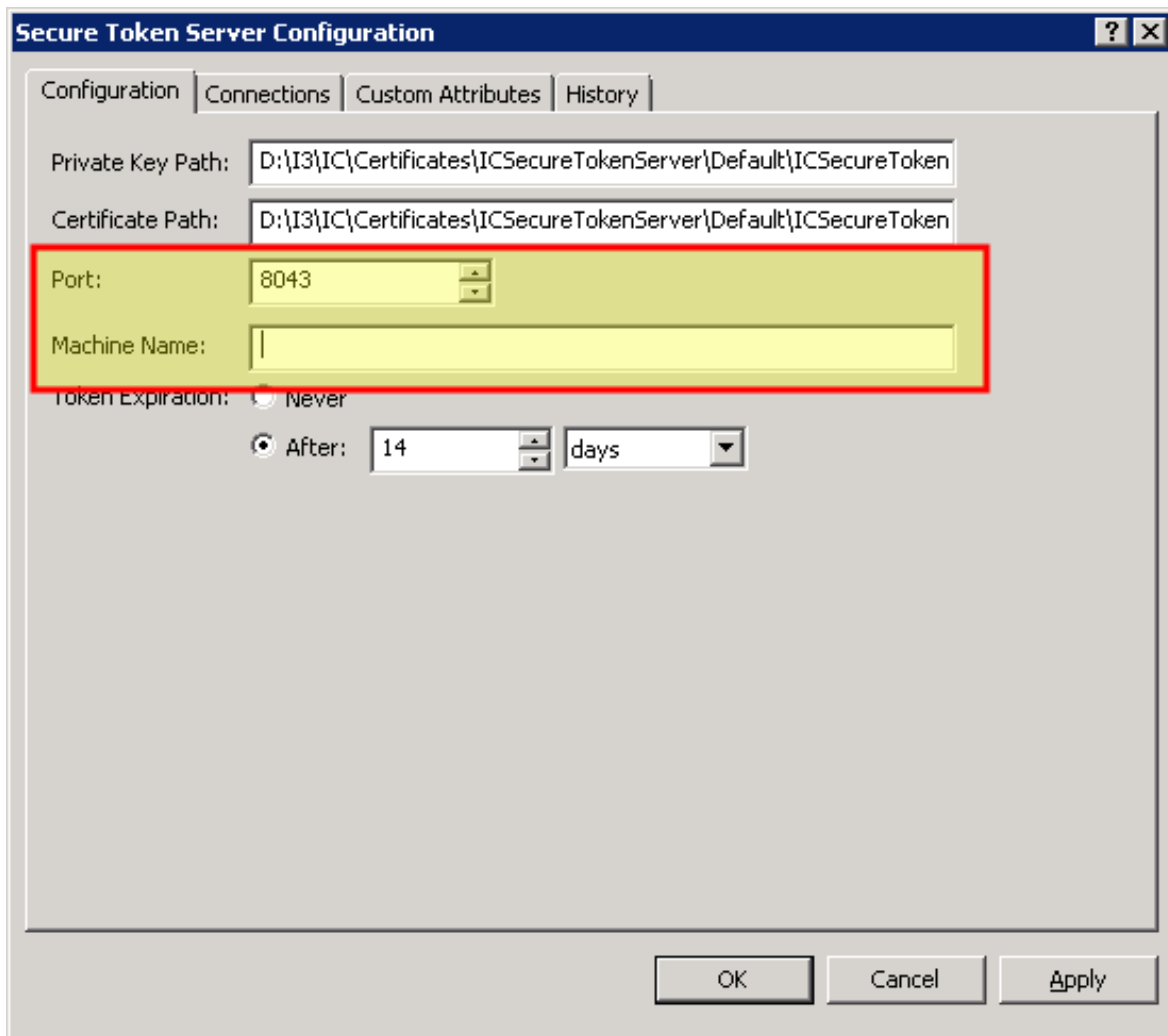
Item	Description
Machine name	The machine name represents the address through which user agents (client applications) can contact the Secure Token Server subsystem of the CIC server. For more information about Secure Token Server addresses, see Single Sign-On configurations .
Network port for HTTPS/SAML messages	The default network port on the CIC server for HTTPS/SAML messages is 8043.

2. Open Interaction Administrator.
3. In the left pane of the **Interaction Administrator** window, expand the **Single Sign-On** container and select the **Secure Token Server** object.
4. In the right pane, double-click the **Configuration** entry.



Interaction Administrator displays the **Secure Token Server Configuration** dialog box.

5. In the **Port** and **Machine Name** controls of the **Secure Token Serve Configuration** dialog box, enter the address through which user agents (CIC client applications) can reach the Secure Token Server subsystem of the CIC server.



Note: If you do not enter text in the **Machine Name** box, the CIC server uses its Fully Qualified Domain Name (FQDN).

Do not enter the port number for an ICWS subsystem (default: 8019) in the **Port** box.

If you are using the CIC server as both the service provider and identity provider, ensure that the address that you enter in the **Machine Name** box is accessible by workstations that host CIC applications for Single Sign-On. This aspect is important if the workstation and CIC server are in different domains.

For more information on the correct address to enter in the **Machine Name** box, see [Single Sign-On configurations](#).

6. If needed, you can change the period of time that can elapse before require re-authentication in the **Token Expiration** control group.
7. Select the **OK** button to save this change and close the **Secure Token Server Configuration** dialog box.

Administer HTTPS certificate for the CIC service provider

By default, when a CIC server starts, it searches for its HTTPS certificate. If it does not find the HTTPS certificate, the CIC server creates it using its Fully Qualified Domain Name as its address in the certificate.

Important!

If the CIC client applications—acting as Single Sign-On user agents—will use an address other than the FQDN of the CIC server to access its resources, you must manually generate a new HTTPS certificate with the non-FQDN address using the GenSSLCertsU.exe command line utility on the CIC server.

[Generate a self-signed HTTPS certificate for non-FQDN configurations](#)

[Generate a non-signing HTTPS certificate for FQDN configurations](#)

[Generate a non-signing HTTPS certificate for non-FQDN configurations](#)

[Replace the CIC HTTPS certificate with an externally-generated certificate](#)

Generate a self-signed HTTPS certificate for non-FQDN configurations

If the workstations hosting CIC client applications that will use Single Sign-On do not use the Fully Qualified Domain Name (FQDN) of the CIC server to access resources, such as using an IP address or short host name, you must use the GenSSLCertsU.exe command line utility to generate the HTTPS certificate for the CIC server.

Important!

You do not need to do this procedure if the workstations hosting your CIC client applications access the CIC server through its Fully Qualified Domain Name (FQDN). The CIC server automatically generates an HTTPS certificate with its FQDN address. Examples of configurations that require you to generate a new HTTPS certificate would be using DNS-A/AAAA records for a switchover pair, IP address, or short host name.

The GenSSLCertsU.exe command line utility has additional switches to customize creation of HTTPS certificates. For more information about the GenSSLCertsU.exe command line utility, see *Security Features Technical Reference*.

1. On the CIC server, open a **Command Prompt** window with Administrator privileges.
2. In the **Command Prompt** window, navigate to the drive where the CIC server software was installed by entering and executing the following command:

D:

D: is the default drive on which the CIC server software is installed. If you installed the CIC server software to a different drive, replace D with the appropriate letter.

3. Navigate to the HTTPS subdirectory by entering and executing the following command:

```
cd \I3\IC\Certificates\HTTPS
```

4. In the **Command Prompt** window, copy the existing files to new, renamed instances by executing the following command:

```
copy CICServerName*.* ?*.*.backup
```

CICServerName is a variable representing the non-FQDN address of this CIC server that CIC client applications in the network can reach. The following table displays examples of different address schemes:

Address scheme	Example
DNS A/AAAA-record	cic-serv.example.com (DNS A/AAAA record points to either cic-serv1.example.com or cic-serv2.example.com)
IP address	192.168.1.100
Short host name	cic-serv

5. In the **Command Prompt** window, enter and execute the following command:

```
GenSSLCertsU w CICServerName -h
```

CICServerName is the non-FQDN address of the CIC server.

The GenSSLCertsU.exe utility generates the HTTPS certificate for the CIC server in the following directory on the partition or hard drive where the CIC server software was installed:

Important!

GenSSLCertsU.exe creates the file name of HTTPS certificate with the host identified with the command. For example, if you have a CIC server named cic-serv1.example.com and you generate a new HTTPS certificate for cic-serv1 (not FQDN), the file names of the certificate and associated files are as follows:

- cic-serv1_Certificate.cer
- cic-serv1_PrivateKey.bin
- cic-serv1_PublicKey.bin
- cic-serv1_TrustedCertificate.cer

The CIC server loads the HTTPS certificate with a file name containing the FQDN of the server, such as cic-serv1.example.com. As such, you must manually rename the newly-generated HTTPS certificate to reflect the FQDN of the CIC server:

- cic-serv1.example.com_Certificate.cer
- cic-serv1.example.com_PrivateKey.bin
- cic-serv1.example.com_PublicKey.bin
- cic-serv1.example.com_TrustedCertificate.cer

The following steps guide you through renaming these files.

6. In the **Command Prompt** window, rename the files for the certificates, PublicKey, and PrivateKey using the following commands:

```
ren CICServerNameCertificate.cer CIC_server_FQDN_Certificate.cer
```

```
ren CICServerNameTrustedCertificate.cer CIC_server_FQDN_TrustedCertificate.cer
```

```
ren CICServerNamePublicKey.bin CIC_server_FQDN_PublicKey.bin
```

```
ren CICServerNamePrivateKey.bin CIC_server_FQDN_PrivateKey.bin
```

CICServerName is a variable that represents the name that the GenSSLCertsU.exe command was given to embed within the HTTPS certificate.

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

Important!

If you use DNS A/AAAA records for a switchover pair, do not replace *CIC_server_FQDN* with the DNS A/AAAA record name for as the target file name. The target file name must be the specific FQDN of this CIC server so that the CIC server loads the certificate and keys automatically.

Also, ensure that you do not miss including the underscore character (_) in the target file name between *CIC_server_FQDN* and the remainder of the file name.

Important!

If you use DNS A/AAAA records for a switchover pair, do not replace *CIC_server_FQDN* with the DNS A/AAAA record name for as the target file name. The target file name must be the specific FQDN of this CIC server so that the CIC server loads the certificate and keys automatically.

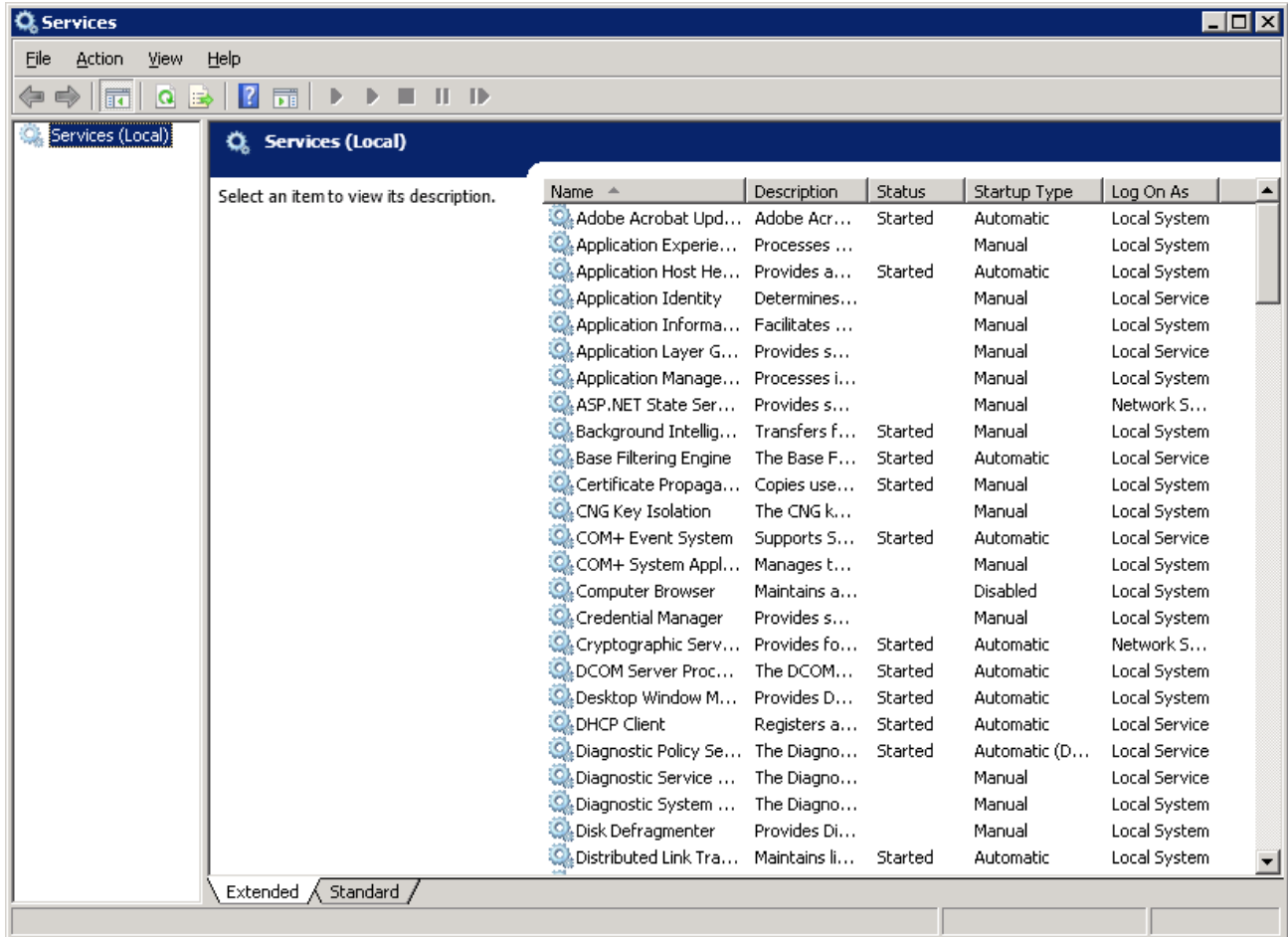
Also, ensure that you do not miss including the underscore character (_) in the target file name between *CIC_server_FQDN* and the remainder of the file name.

Examples:

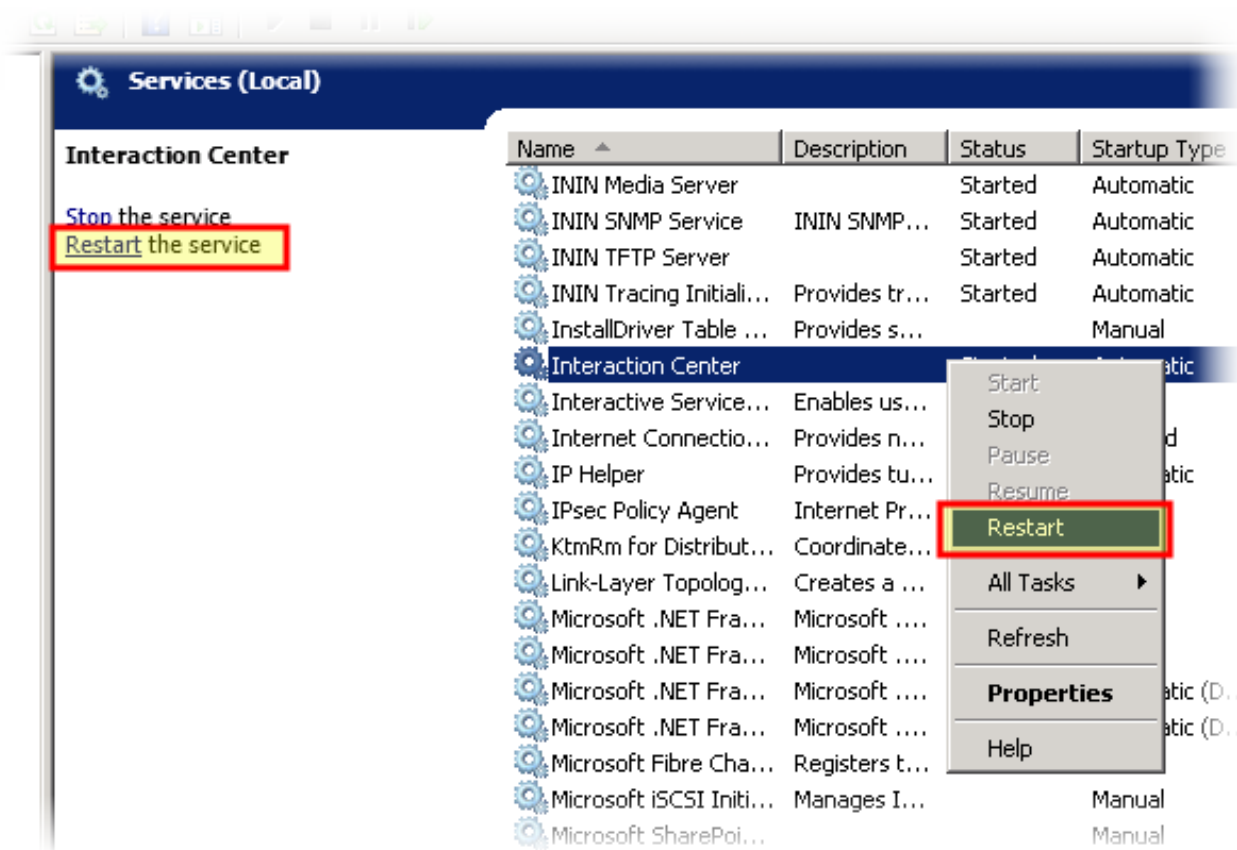
```
ren cic-serv1Certificate.cer cic-serv1.example.com_Certificate.cer
ren cic-serv1TrustedCertificate.cer cic-serv1.example.com_TrustedCertificate.cer
ren cic-serv1PublicKey.bin cic-serv1.example.com_PublicKey.bin
```

```
ren cic-serv1PrivateKey.bin cic-serv1.example.com_PrivateKey.bin
```

7. In the Windows Control Panel, start the **Services** application.



8. Restart the **Interaction Center** service by right-clicking it and selecting **Restart** from the resulting context menu or selecting the **Restart** hyperlink on the left side of the pane.



The CIC server restarts and uses the new HTTPS certificate that you generated.

Generate a non-signing HTTPS certificate for FQDN configurations

Important!

Do not do this procedure unless you use a Fully Qualified Domain Name (FQDN) for your CIC server (service provider) and your identity provider requires your SAML SSO messages to use a non-signing HTTPS certificate.

Some identity providers, such as IBM LightHouse, require a non-signing HTTPS certificate. If your identity provider requires this type of HTTPS certificate and your CIC client workstations use FQDN to reach the CIC server (service provider), do the following steps:

1. On the CIC server, open a **Command Prompt** window with Administrator privileges.
2. In the **Command Prompt** window, navigate to the drive where the CIC server software was installed by entering and executing the following command:

D:

D: is the default drive on which the CIC server software is installed. If you installed the CIC server software to a different drive, replace D with the appropriate letter.

3. Navigate to the Lines subdirectory by entering and executing the following command:
cd \I3\IC\Certificates\Lines

4. In the **Command Prompt** window, copy the existing files to new, renamed instances by executing the following command:

```
copy CIC_server_FQDN*. * ?*. *.backup
```

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

5. In the **Command Prompt** window, enter and execute the following command:

```
GenSSLCertsU I CIC_server_FQDN
```

CIC_server_FQDN is the address of the CIC server that can be reached by CIC client applications in the network.

The GenSSLCertsU.exe utility generates the HTTPS certificate in the \I3\IC\Certificates\Lines directory on the partition or hard drive where the CIC server software was installed.

6. Navigate to the DefaultLinesAuthority directory by executing the following command:

```
cd \I3\IC\Certificates\LinesAuthority\DefaultLinesAuthority\
```

7. Copy and rename the LinesAuthority.cer file to the \I3\IC\Certificates\HTTPS directory by executing the following command:

```
copy LinesAuthority.cer \I3\IC\Certificates\HTTPS\CIC_server_FQDN_TrustedCertificate.cer
```

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

Important!

Ensure that you include the underscore character (_) between the FQDN address of the CIC server and the remainder of the target file name.

8. Navigate to the HTTPS directory by executing the following command:

```
cd \I3\IC\Certificates\HTTPS
```

9. If files for this CIC server address already exist in this directory, rename the files by executing the following command in the **Command Prompt** window:

10. **ren *CIC_server_FQDN**.* ?*.*.backup**

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

Important!

Ensure that you do not rename the *CIC_server_FQDN_TrustedCertificate.cer* file that you created in this directory in step 7.

11. Copy the files in the \I3\IC\Certificates\Lines directory to the \I3\IC\Certificates\HTTPS directory by executing the following command:

```
copy ..\Lines\CIC_server_FQDN* .
```

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

Important!

Ensure to include the space between the asterisk (*) and the period (.) in the copy command.

12. Use the ren command to rename each of these files so that an underscore character (_) is between the CIC server name and the remainder of the filename:

```
ren CIC_server_FQDNCertificate.cer CIC_server_FQDN_Certificate.cer
```

```
ren CIC_server_FQDNPrivateKey.bin CIC_server_FQDN_PrivateKey.bin
```

```
ren CIC_server_FQDNPublicKey.bin CIC_server_FQDN_PublicKey.bin
```

CIC_server_FQDN is a variable representing the FQDN of this CIC server.

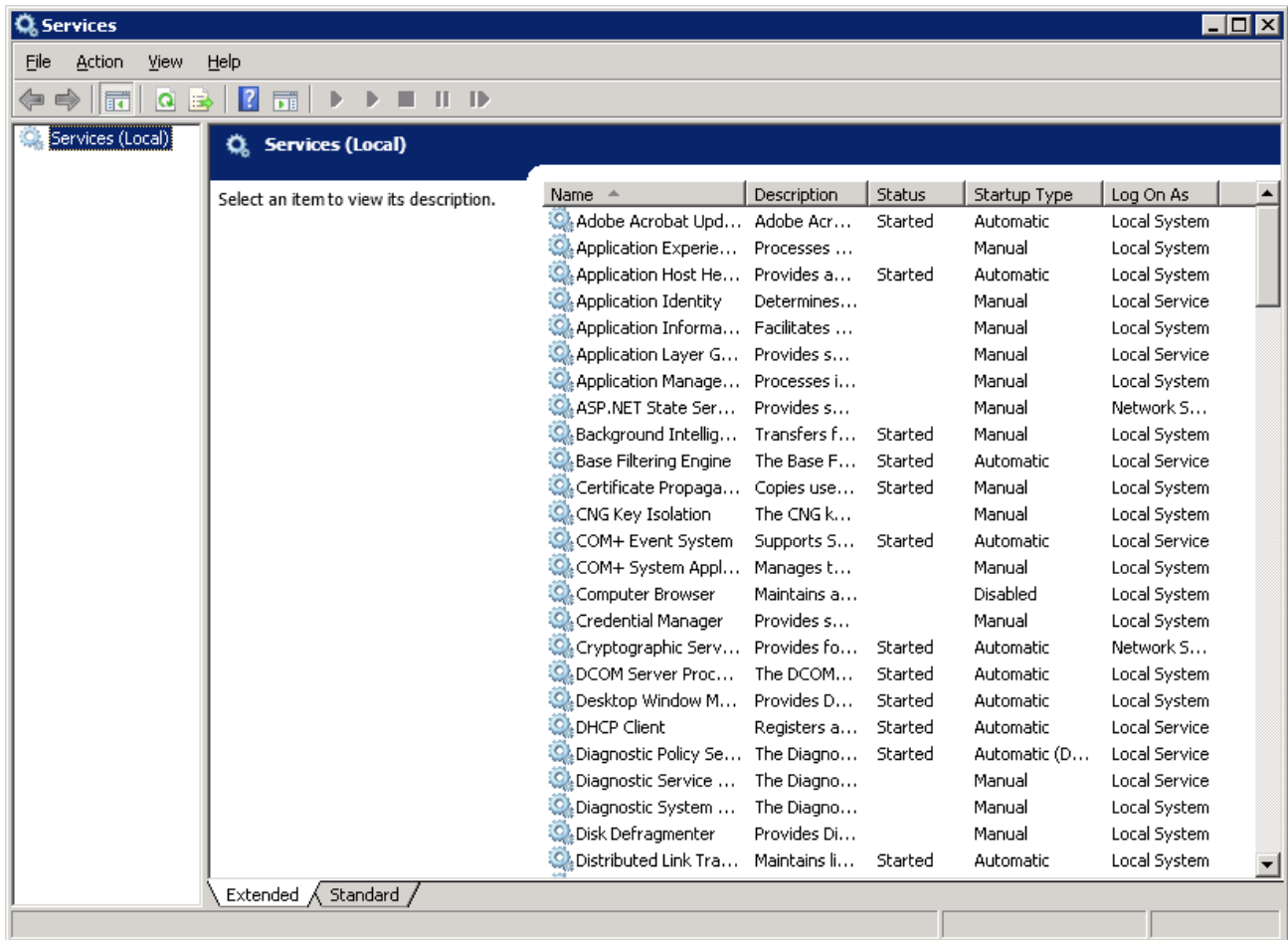
Examples:

```
ren cic-serv1.example.comCertificate.com cic-serv1.example.com_Certificate.cer
```

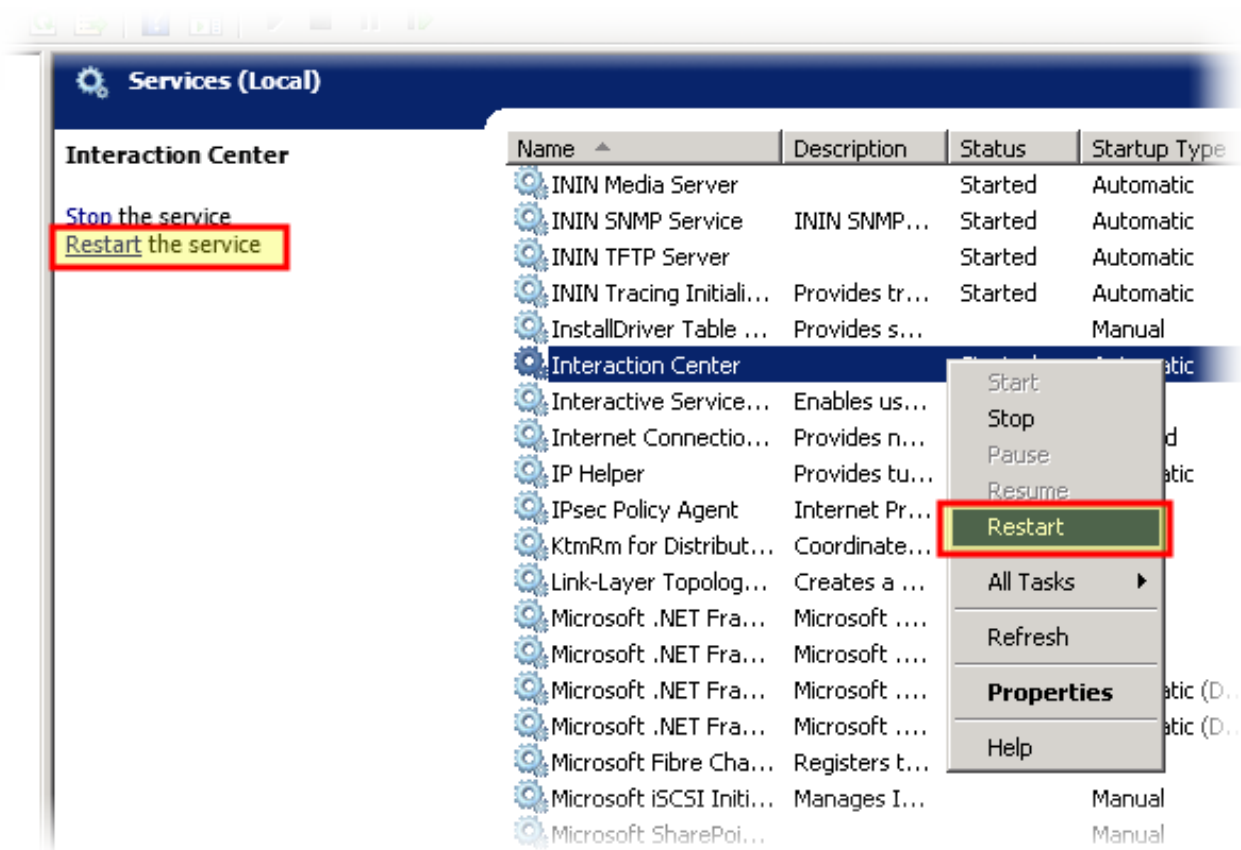
```
ren cic-serv1.example.comPrivateKey.bin cic-serv1.example.com_PrivateKey.bin
```

```
ren cic-serv1.example.comPublicKey.bin cic-serv1.example.com_PublicKey.bin
```

13. In the Windows **Control Panel**, start the **Services** application.



- Restart the **Interaction Center** service by right-clicking it and selecting **Restart** from the resulting context menu or selecting the **Restart** hyperlink on the left side of the pane.



The CIC server restarts and uses the new HTTPS certificate that you generated.

Generate a non-signing HTTPS certificate for non-FQDN configurations

Important!

Do not do this procedure unless you use an IP address, short host name, or DNS A/AAAA record for your CIC server (service provider) or switchover pair, and your identity provider requires your SAML SSO messages to use a non-signing HTTPS certificate.

Some identity providers, such as IBM LightHouse, require a non-signing HTTPS certificate. If your identity provider requires this type of HTTPS certificate and your CIC client workstations use an IP address, short host name, or DNS A/AAAA record to reach the CIC server (service provider), do the following steps:

1. On the CIC server, open a **Command Prompt** window with Administrator privileges.
2. In the **Command Prompt** window, navigate to the drive where the CIC server software was installed by entering and executing the following command:

D:

D: is the default drive on which the CIC server software is installed. If you installed the CIC server software to a different drive, replace D with the appropriate letter.

3. Navigate to the Lines subdirectory by entering and executing the following command:

```
cd \I3\IC\Certificates\Lines
```

4. In the **Command Prompt** window, copy the existing files to new, renamed instances by executing the following command:

```
copy CICServerName*.* ?*.*.backup
```

CICServerName is a variable representing the name of this CIC server.

5. In the **Command Prompt** window, enter and execute the following command:

```
GenSSLCertsU | CICServerName
```


CICServerName is the IP address, short host name, or DNS A/AAAA record of the CIC server or switchover pair that can be reached by CIC client applications in the network.

The GenSSLCertsU.exe utility generates the files for the HTTPS certificate, PublicKey, and PrivateKey, and copies them to the \I3\IC\Certificates\Lines directory.

6. Navigate to the DefaultLinesAuthority directory by executing the following command:

```
cd \I3\IC\Certificates\LinesAuthority\DefaultLinesAuthority\
```

7. Copy and rename the LinesAuthority.cer file to the \I3\IC\Certificates\HTTPS directory by entering and executing the following command:

```
copy LinesAuthority.cer \I3\IC\Certificates\HTTPS\CICServerName_TrustedCertificate.cer
```

CICServerName represents a variable for the IP address, short host name, or DNS A/AAAA record of this CIC server or switchover pair.

Important!

Ensure that you include the underscore character (_) between the FQDN address of the CIC server and the remainder of the target file name.

8. Navigate to the HTTPS directory by executing the following command:

```
cd \I3\IC\Certificates\HTTPS
```

9. If files for this CIC server address already exist in this directory, rename the files by executing the following command in the Command Prompt window:

10. **ren *CICServerName**.* ?*.*.backup**

CICServerName is a variable representing the IP address, short host name, or DNS A/AAAA record of this CIC server or switchover pair.

11. Copy the files in the \I3\IC\Certificates\Lines directory to the \I3\IC\Certificates\HTTPS directory by executing the following command:

```
copy ..\Lines\CICServerName* .
```

CICServerName is a variable representing the IP address, short host name, or DNS A/AAAA record of this CIC server or switchover pair.

Important!

Ensure to include the space between the asterisk (*) and the period (.).

12. Use the ren command to rename each of these files so that an underscore character (_) is between the CIC server name and the remainder of the filename:

```
ren CICServerNameCertificate.cer CICServerName_Certificate.cer
```

```
ren CICServerNamePrivateKey.bin CICServerName_PrivateKey.bin
```

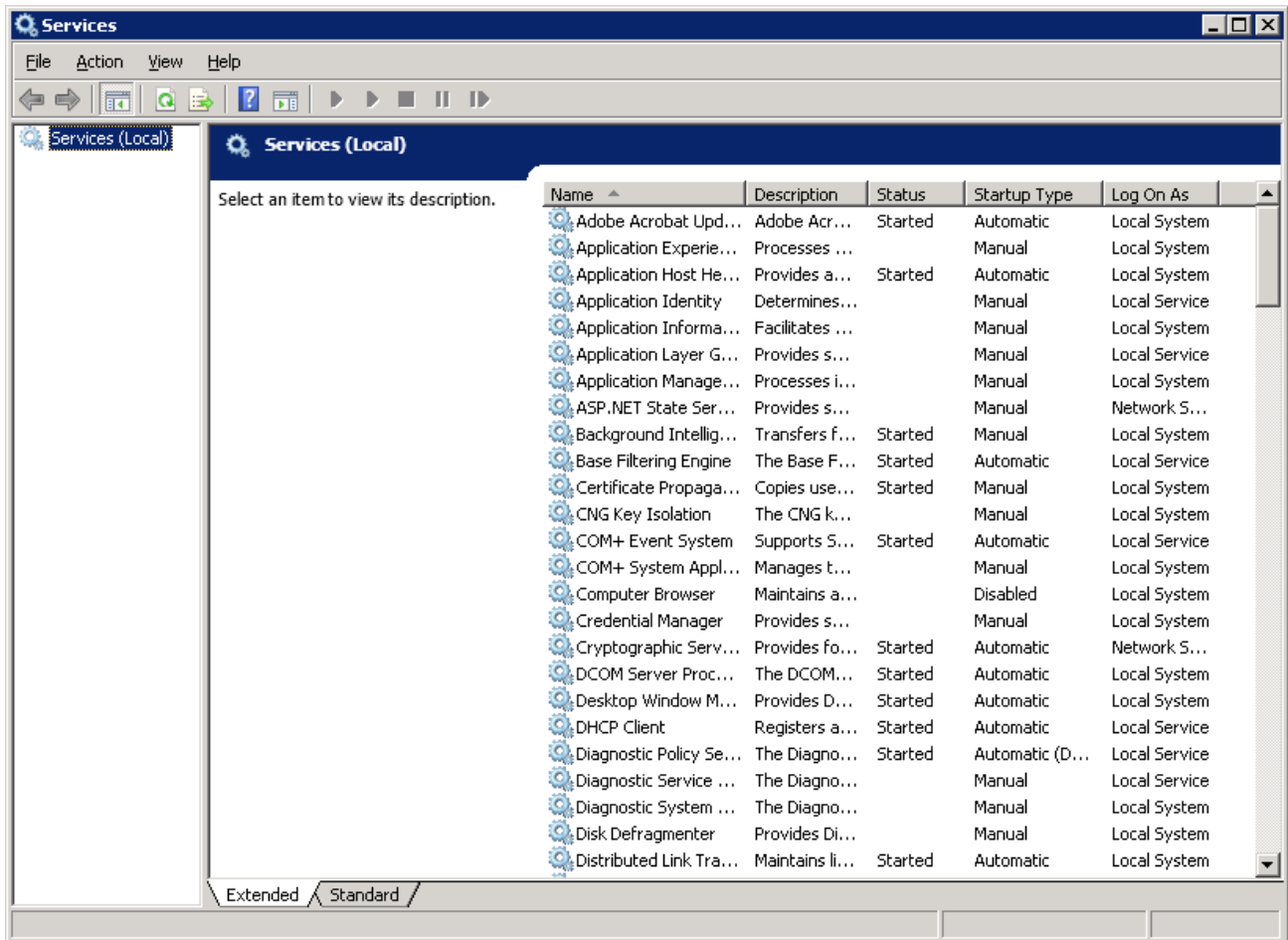
```
ren CICServerNamePublicKey.bin CICServerName_PrivateKey.bin
```

CICServerName is a variable representing the IP address, short host name, or DNS A/AAAA record of this CIC server or switchover pair.

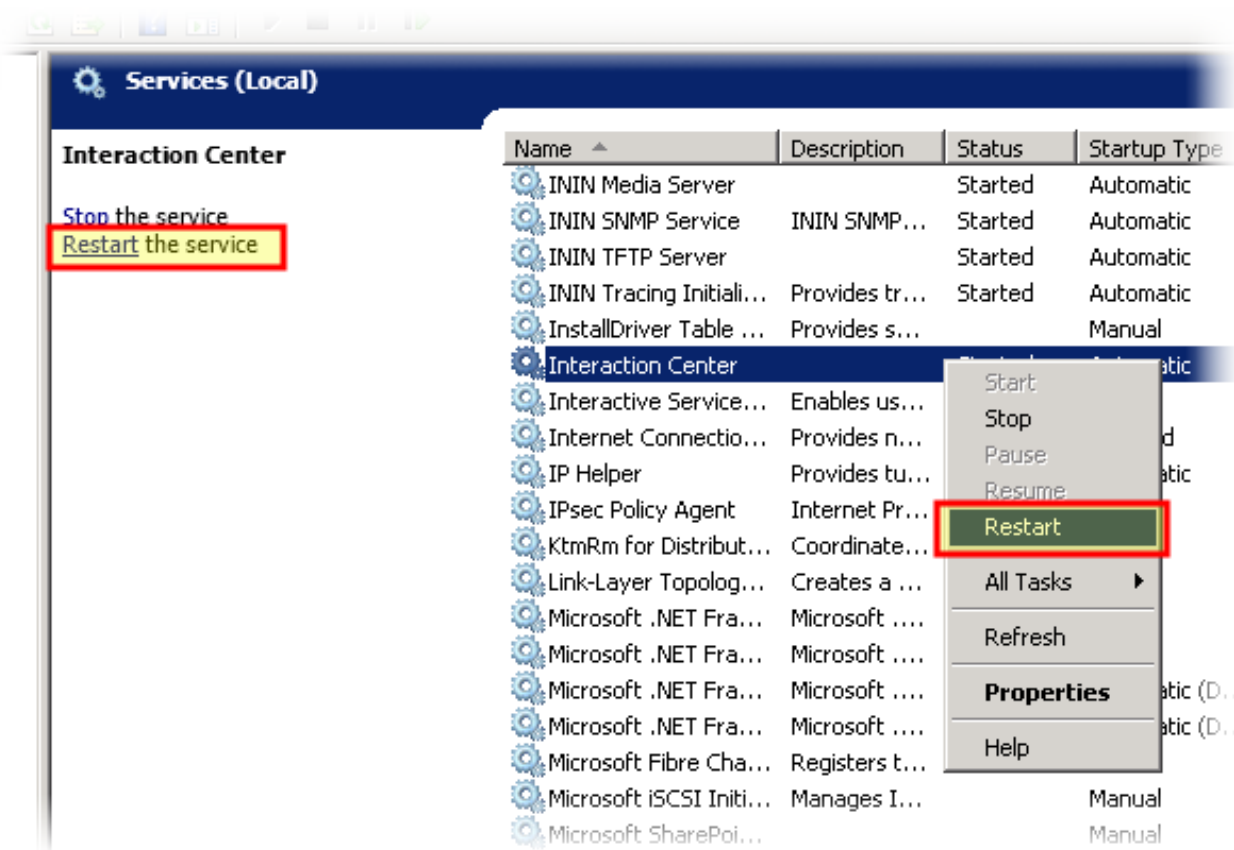
Examples:

```
ren cic-serv1Certificate.cer cic-serv1_Certificate.cer
ren cic-serv1PrivateKey.bin cic-serv1_PrivateKey.bin
ren cic-serv1PublicKeybin cic-serv1_PublicKey.bin
```

13. In the Windows Control Panel, start the Services application.



- Restart the **Interaction Center** service by right-clicking it and selecting **Restart** from the resulting context menu or selecting the **Restart** hyperlink on the left side of the pane.



The CIC server restarts and uses the new HTTPS certificate that you generated.

Replace the CIC HTTPS certificate with an externally-generated certificate

If you have generated your own HTTPS certificate outside of CIC or if you use one generated by a third-party entity, you can use it on the CIC server (service provider).

Third-party certificates and Certificate Authorities

You can use a Certificate Signing Request (CSR) to obtain a signed certificate from a Certificate Authority (CA). The CSR generates a Private Key, which you must keep secure, and a certificate containing a Public Key. To generate a CSR, you must provide the following information:

Organization	Usually, the exact legal name of your company or organizational entity.
Organizational Unit	The department or division of the organization that will use the certificate, such as IT or PureConnect Customer Care.
Common Name	Usually, the Fully Qualified Domain Name of the service that will use the certificate. For devices that do not use a name, use the IP address of the device as the Common Name.

Some other information, such as country, city, state, may be required by the CA to which you submit the CSR. Additional information that could be included in the certificate, such as an e-mail address is optional.

The method through which you generate a CSR and the information required can vary between different CAs. Some CAs provide utilities that you can download to generate the CSR. You can also use other third-party tools, such as OpenSSL or the certreq command on Windows Server.

Note: Genesys does not provide technical support for third-party tools through which you generate a CSR. Consult the CA or vendor of these tools for assistance.

Once you have generate a CSR with a .req file extension, you can submit it to your selected CA. The method through which you submit your CSR is dependent on your selected CA. When the CA returns the signed certificate, you can install it on your CIC server.

An advantage of using third-party or self-generated certificates is that you do not have to install the certificate that is automatically generated by the CIC server into all client workstations. Client workstations can validate the CA signed certificate through existing entries in their own root certificate stores.

Procedure

After you have received your signed certificate from your selected Certificate Authority, or if you have generated your own certificate for use, do the following steps to install the certificate on the CIC server for Single Sign-On operations:

Important!

If the certificate was generated on another Windows server, you must create a .PFX (PKCS#12 format) file, which contains the Private Key, the Public Key, and the certificate (signed container for the public key) so that you can import it into another Windows server, such as the CIC server. You can use various tools to generate the .PFX file, including the Microsoft Management Console Certificates snap-in, OpenSSL, and the `ssl_app-w32r-5-0.exe` file in the `\I3\IC\Server` directory.

1. Copy the .PFX file to the CIC Server.
2. On the CIC server, open a **Command Prompt** window with Administrator privileges.
3. If necessary, change to the drive where the CIC server software was installed by entering and executing the following command:

DriveLetter.

DriveLetter is the letter representing the partition or hard drive where the CIC server software was installed. By default, the CIC server software is installed on the D: drive.

4. Navigate to the `\I3\IC\Server` directory using the `cd` command as follows:

```
cd \I3\IC\Server
```

5. Export the certificate from the .PFX file using the following command:

```
ssl_app-w32r-5-0.exe pkcs12 -in PFXFile.pfx -nokeys -out PFXCertificateFile_Certificate.cer -nodes
```

PFXFile is a variable representing the drive, directory path, and file name of the .PFX file that you copied to the CIC server. Example: `D:\MyPFXFile.pfx`

PFXCertificateFile is a variable representing the FQDN of the CIC server. Example: `cic-serv1.example.com_Certificate.cer`

6. If you use non-self-signed certificates (there are additional certificates in the chain), do the following additional steps:
 - a. Execute the following command to export the CA certificate file in DER format:

```
ssl_app-w32r-5-0.exe pkcs12 -in PFXFile.pfx -nokeys -out PFXCACertificateFile_Certificate.cer -nodes -cacerts
```

PFXFile is a variable representing the drive, directory path, and file name of the .PFX file that you copied to the CIC server. Example: `D:\MyPFXFile.pfx`

PFXCACertificateFile is a variable that you can name freely.

- b. Execute the following command to convert the exported CA certificate to x509 (PEM) format:

```
ssl_app-w32r-5-0.exe x509 -in PFXCACertificate.cer -out CACertificate.cer
```

PFXCACertificate is a variable representing the drive, directory path, and file name that you chose in step a.

CACertificate is a variable that you can name freely.

- c. Using a text editor, open the CA certificate (*CACertificate.cer*) that you exported in step a.
 - d. Copy all of the text to the Windows clipboard by pressing the **Ctrl+C** key combination on your keyboard.
 - e. In the text editor, open the certificate (*PFXCertificateFile_Certificate.cer*) that you exported in step 5.
 - f. Place the cursor at the end of the contents of the certificate and press the **Ctrl+V** key combination to append the contents of the CA certificate from the Windows clipboard.
 - g. Save the certificate file (*PFXCertificateFile_Certificate.cer*) and close the text editor.
7. Export the Private Key from the .PFX file by executing the following command:

```
ssl_app-w32r-5-0.exe pkcs12 -in PFXFile.pfx -nocerts -out PFXPrivateKey_PrivateKey.bin -nodes
```

PFXFile is a variable representing the drive, directory path, and file name of the .PFX file that you copied to the CIC server. Example: `D:\MyPFXFile.pfx`

PFXPrivateKey is a variable that you can name freely.

- Convert the Private Key that you exported from the .PFX file to the RSA format by executing the following command:

```
ssl_app-w32r-5-0.exe rsa -in PFXPrivateKey.bin -out PrivateKey_PrivateKey.bin
```

PFXPrivateKey is a variable representing the name you specified in step 7.

PrivateKey is a variable representing the FQDN of the CIC server.

Example: cic-serv1.example.com_PrivateKey.bin

Important!

The file name of the Private Key (.bin) must match the file name of the certificate (.cer) that you exported in step 5.

- Export the Public Key from the Private Key by executing the following command:

```
ssl_app-w32r-5-0.exe rsa -in PrivateKey_PrivateKey.bin -out PublicKey_PublicKey.bin -RSAPublicKey_out
```

PrivateKey is a variable representing the FQDN of the CIC server. Example: cic-serv1.example.com_PrivateKey.bin

PublicKey is a variable representing the FQDN of the CIC server. Example: cic-serv1.example.com_PublicKey.com

- In the \I3\IC\Certificates\HTTPS directory, make backup copies (in a different directory) of any existing certificates.
- Copy the following files to the \I3\IC\Certificates\HTTPS directory:

- CIC_server_FQDN_Certificate*.cer

Example: cic-serv1.example.com_Certificate.cer

- CIC_server_FQDN_PrivateKey*.bin

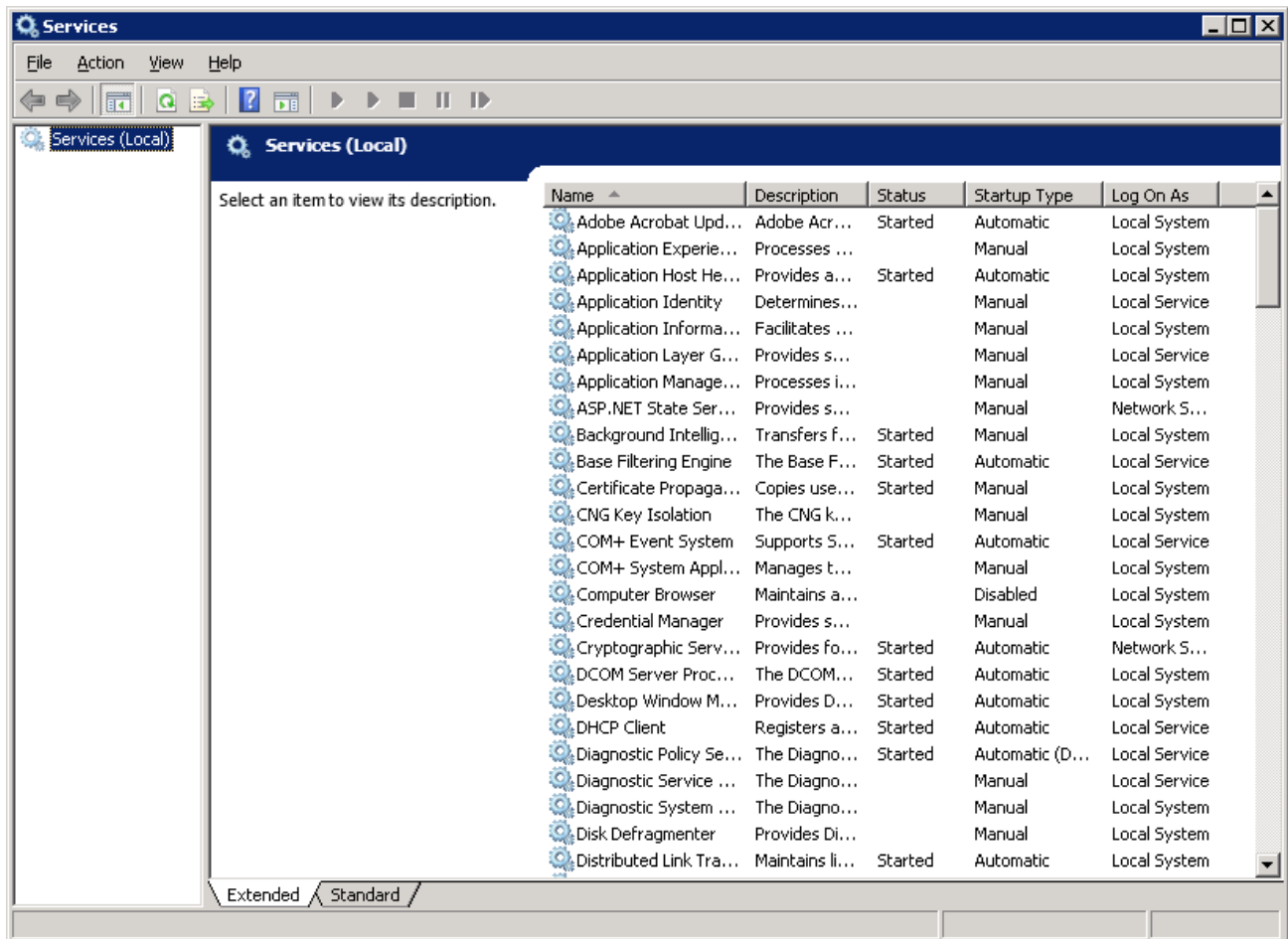
Example: cic-serv1.example.com_PrivateKey.bin

- CIC_server_FQDN_PublicKey*.bin

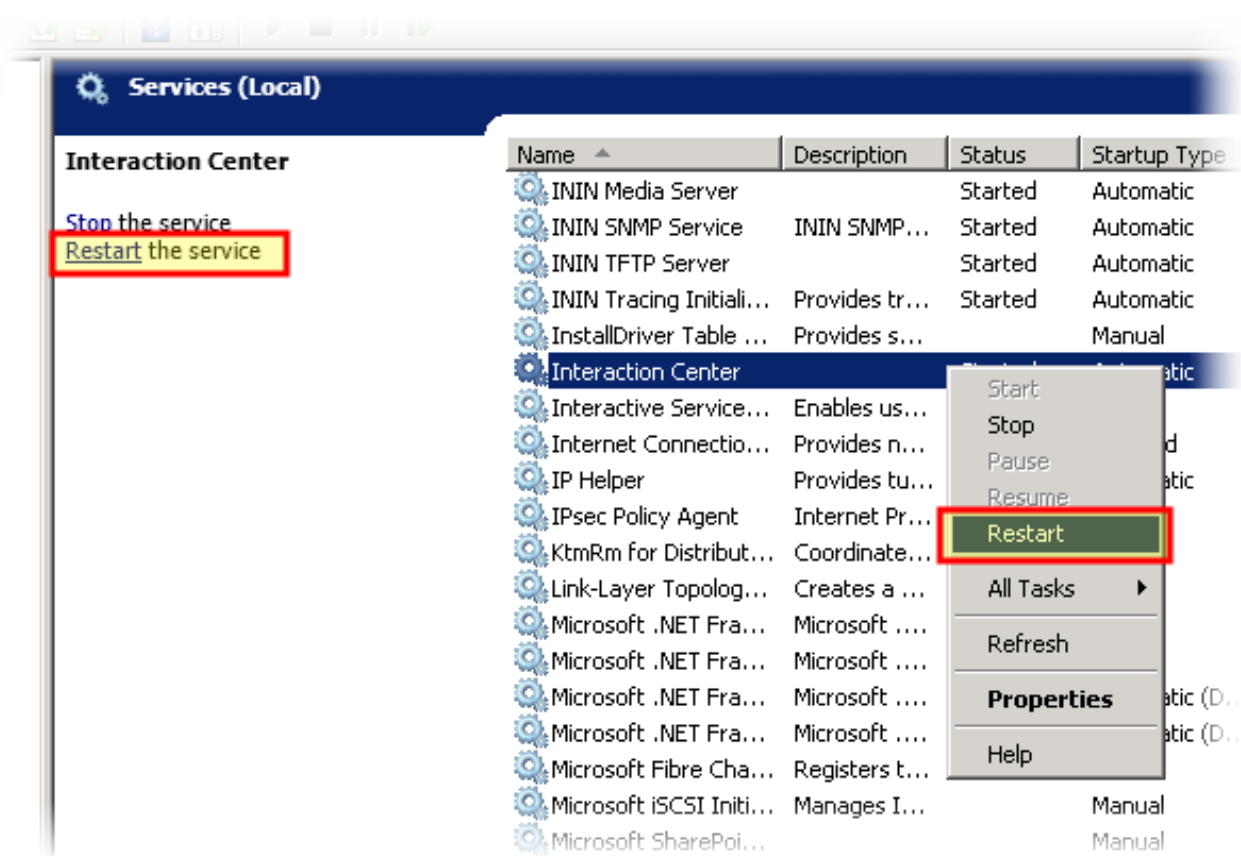
Example: cic-serv1.example.com_PublicKey.bin

CIC_server_FQDN is a variable representing the FQDN of the CIC server.

- In the Windows Control Panel, start the Services application.



- Restart the **Interaction Center** service by right-clicking it and selecting **Restart** from the resulting context menu or selecting the **Restart** hyperlink on the left side of the pane.



The CIC server restarts and uses the new HTTPS certificate that you exported.

Configure identity provider settings for CIC

This section contains the following procedures:

[Copy identity provider validation certificates to the CIC server](#)

[Ensure the format of the validation certificates](#)

[Configure identity provider settings in Interaction Administrator](#)

[Import SAML 2.0 metadata from identity provider](#)

[Manually configure identity provider settings](#)

Copy identity provider validation certificates to the CIC server

In [Gather identity provider information](#), you collected one or more validation certificates from your selected identity provider. You must now create a directory on the CIC server and copy those validation certificates to that location.

1. On the CIC server, open a **Command Prompt** window as an Administrator.
2. If necessary, change to the drive where the CIC server software was installed by entering and executing the following command:

DriveLetter.

DriveLetter is the letter representing the partition or hard drive where the CIC server software was installed. By default, the CIC server software is installed on the D: drive.

3. Use the **md** command to create a new directory to store the validation certificates of the identity provider.

Example:

```
md \I3\IC\Certificates\ICSecureTokenServer\ValidationCertificates\IdentityProviderName
```

IdentityProviderName is a variable representing a directory name that you create for this identity provider.

4. Use the **copy** command to copy the validation certificates of the identity provider from a resource or network location to this directory.

Examples:

```
copy U:\*.cer D:\I3\IC\Certificates\ICSecureTokenServer\ValidationCertificates\IdentityProvider1
```

```
copy \\myexamplepc\share1\*.cer
```

```
D:\I3\IC\Certificates\ICSecureTokenServer\ValidationCertificates\IdentityProvider1
```

Important!

If you are using a switchover pair of CIC servers, copy the validation certificate to both the primary and secondary servers. The path and file name that you create for the validation certificate must be identical on both servers for Single Sign-On to remain functional after a switchover event.

Ensure the format of the validation certificates

Validation certificates from the identity provider that you will use in your CIC Single Sign-On environment must be valid and formatted in PEM (base-64) format.

1. On the CIC server, open a **Command Prompt** window with Administrator privileges.
2. If necessary, change to the drive where the CIC server software was installed by entering and executing the following command:

DriveLetter.

DriveLetter is the letter representing the partition or hard drive where the CIC server software was installed. By default, the CIC server software is installed on the D: drive.

3. Enter and execute the following command:

```
\I3\IC\Server\ssl_app-w32r-5-0.exe x509 -inform PEM -in IdentityProviderName -text
```

IdentityProviderName is a variable representing the local directory on the CIC server and the file name to which you copied the validation certificate of the identity provider. You created this directory in step 3 of the [Copy identity provider validation certificates to the CIC server](#) procedure.

If you can read field names, such as **Version**, in the output from the command, the validation certificate is in PEM format and is valid. You can continue to the next procedure: [Configure identity provider settings in Interaction Administrator](#). Otherwise, the validation certificate is likely in DER format. Continue to the next step to convert the certificate into PEM format.

4. In the **Command Prompt** window, enter and execute the following command:

```
start explorer /e,PathToValidationCertificate
```

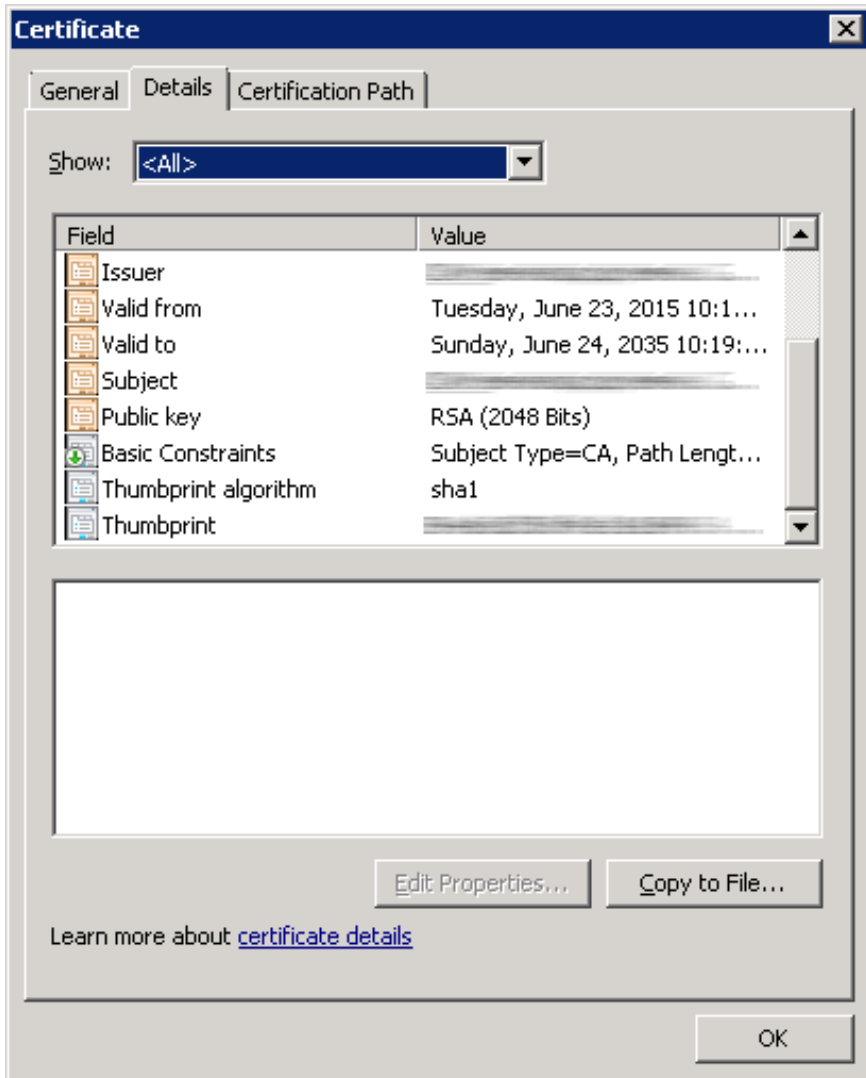
PathToValidationCertificate is a variable representing the location directory on the CIC to which you copied the validation certificate of the identity provider.

Example:

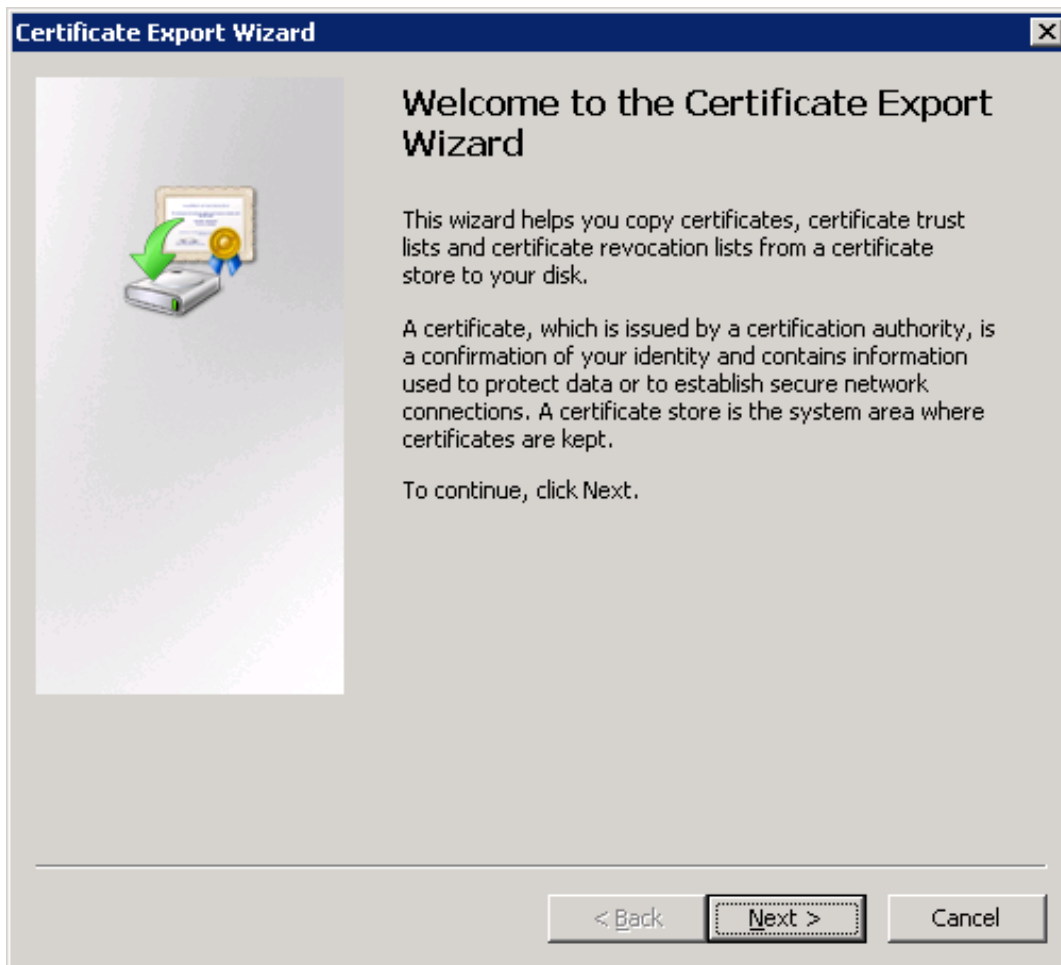
```
start explorer /e,D:\I3\IC\Certificates\ICSecureTokenServer\ValidationCertificates\IdentityProvider1
```

Windows opens an Explorer window displaying the contents of the specified directory.

- In the Windows Explorer window, locate the validation certificate of the identity provider and double-click it. Windows displays the **Certificate** dialog box with information for the validation certificate.
- In the **Certificate** dialog box, select the **Details** tab.

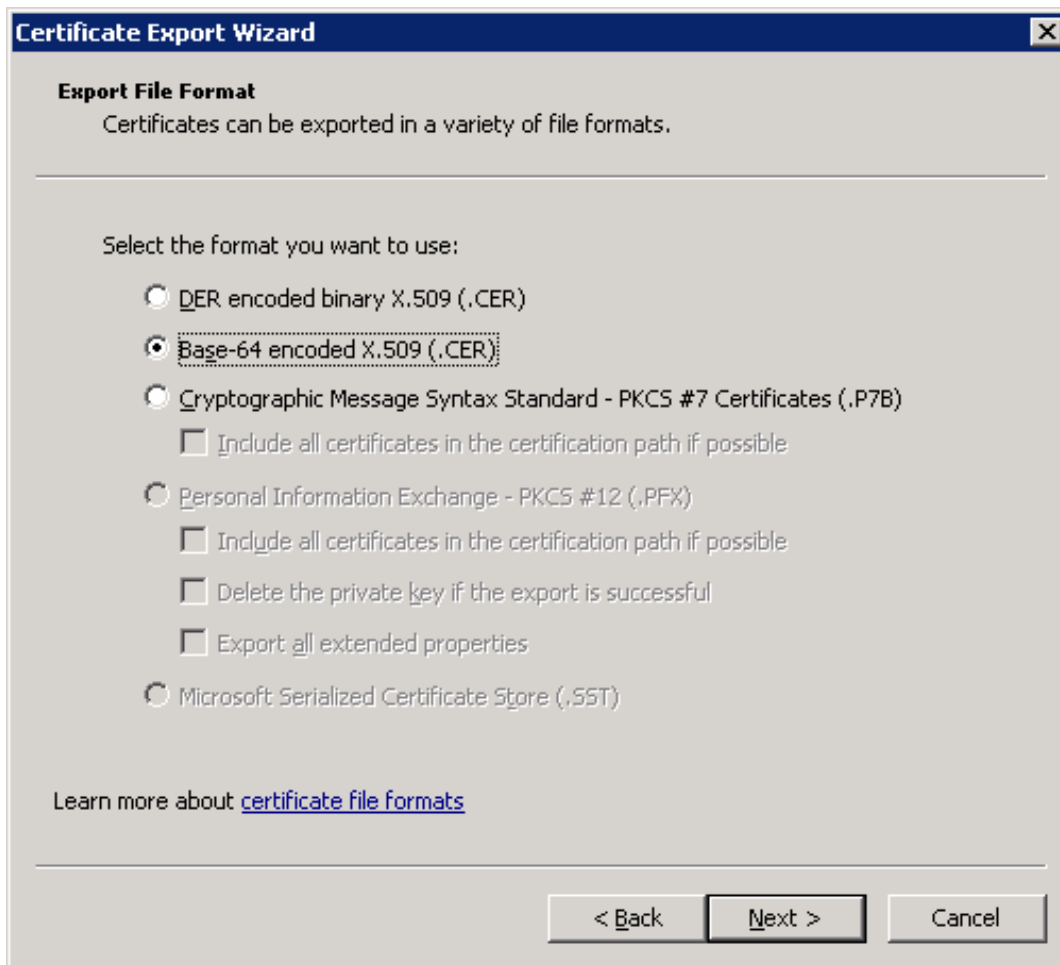


- Select the **Copy to File** button. Windows displays the **Certificate Export Wizard** dialog box.

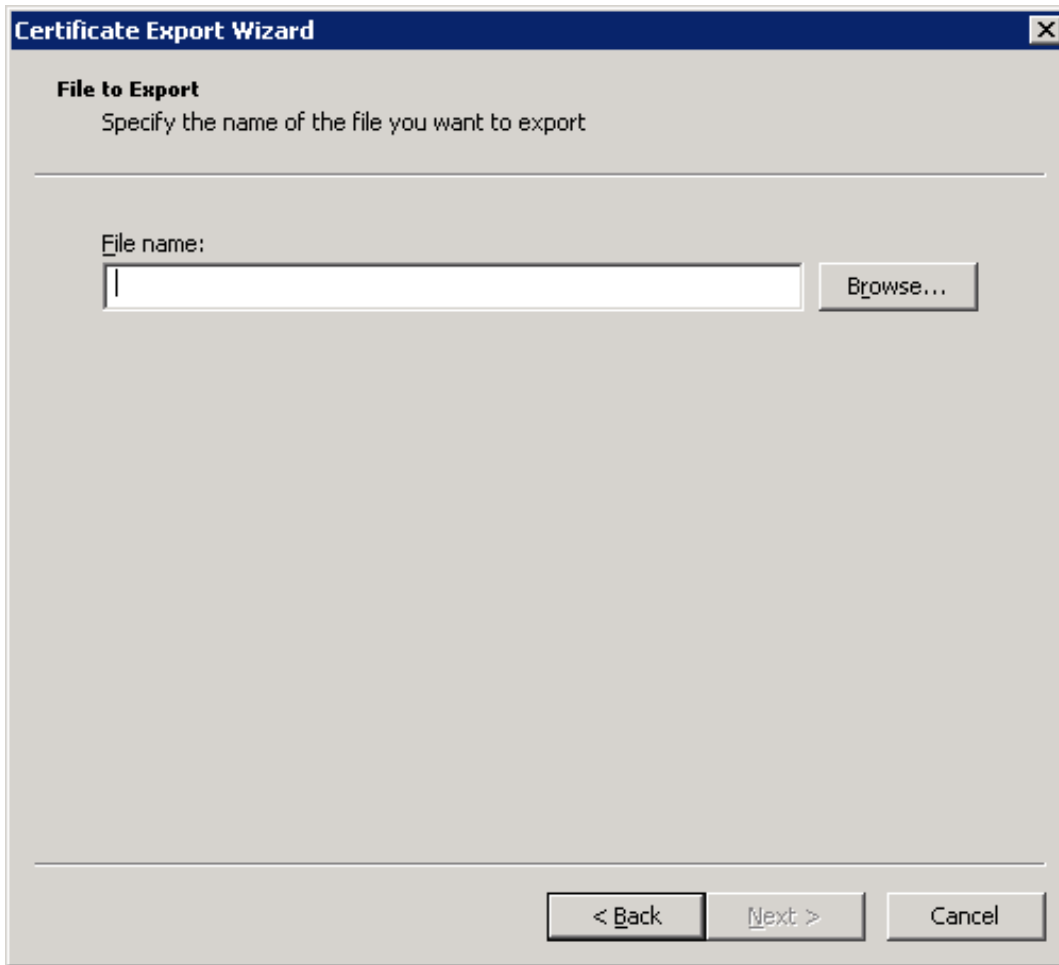


8. Select the **Next** button.

Certificate Export Wizard displays the **Export File Format** screen.



9. Select the **Base-64 encoded X.509 (.CER)** option.
10. Select the **Next** button.
Certificate Export Wizard displays the **File to Export** screen.

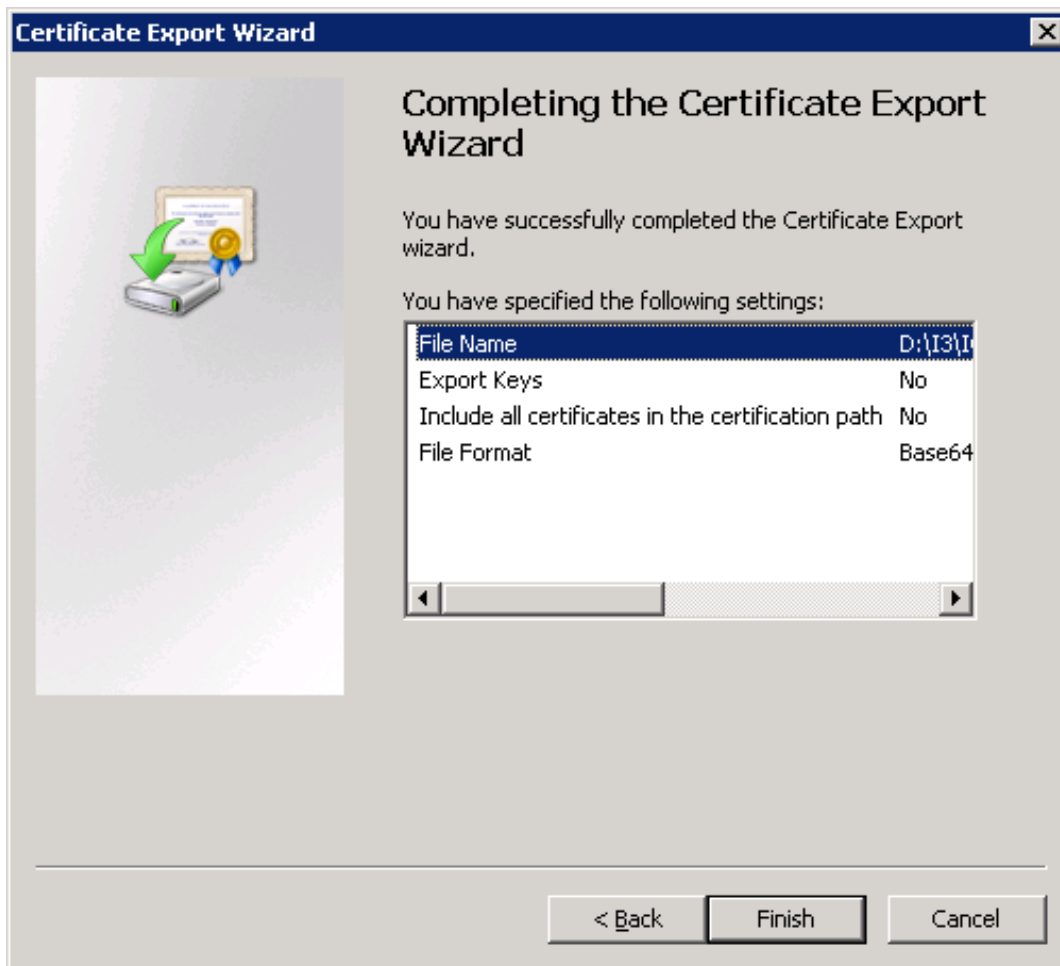


11. In the **File name** box, enter the path and file name that you want to create as the PEM format of the validation certificate for the identity provider.

Tip: You can use the **Browse** button to navigate the existing file system and specify a file name.

12. Select the **Next** button.

Certificate Export Wizard displays the final screen.



13. Select the **Finish** button.
14. In the **Certificates** dialog box, select the **OK** button to close it.

Your certificate is now in the PEM format.

Configure identity provider settings in Interaction Administrator

Customer Interaction Center (CIC) provides the following methods for configuring identity provider settings on the CIC server:

- [Import SAML 2.0 metadata from identity provider](#) - Some identity providers can produce a SAML 2.0 metadata file that contains some of the settings that the CIC server requires for the configuration of a Single Sign-On profile and binding for that identity provider. In Interaction Administrator, you can import that file through a wizard interface. This method is efficient and less prone to errors.
- [Manually configure identity provider settings](#) - If the identity provider does not produce a SAML 2.0 settings file, you can manually configure the CIC server through Interaction Administrator.

Import SAML 2.0 metadata from identity provider

The SAML 2.0 metadata file from the identity provider contains metadata in an XML format. Interaction Administrator can import this metadata and automatically configure some CIC server Single Sign-On settings for that identity provider.

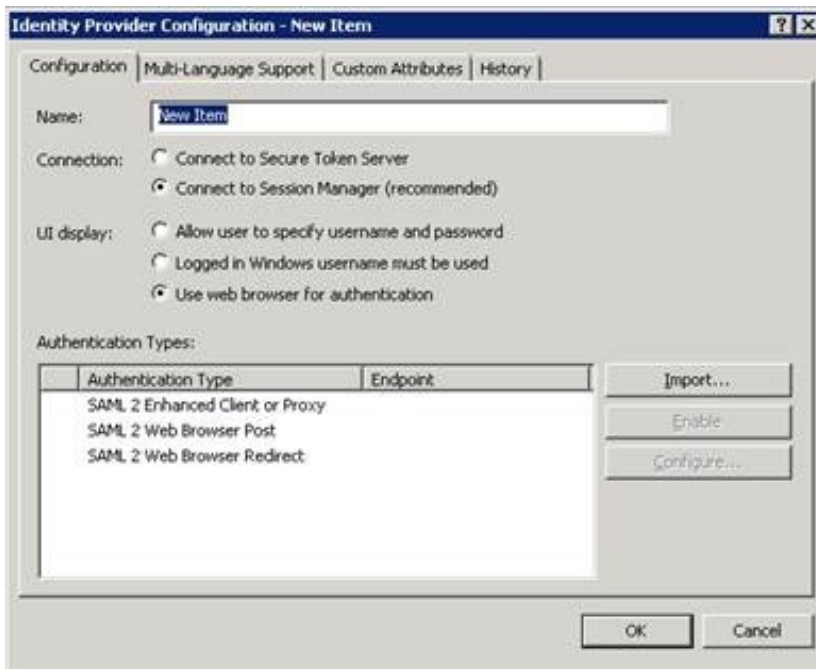
Important!

The SAML 2.0 metadata file does not contain *claims* that the identity provider returns with SAML <AuthnRequest> responses. You must have the claims from the identity provider so that you can enter them in the wizard interface. You should have already gathered the claims in [Gather identity provider information](#).

1. Open Interaction Administrator for the CIC server for which you want to configure Single Sign-On settings for the identity provider.

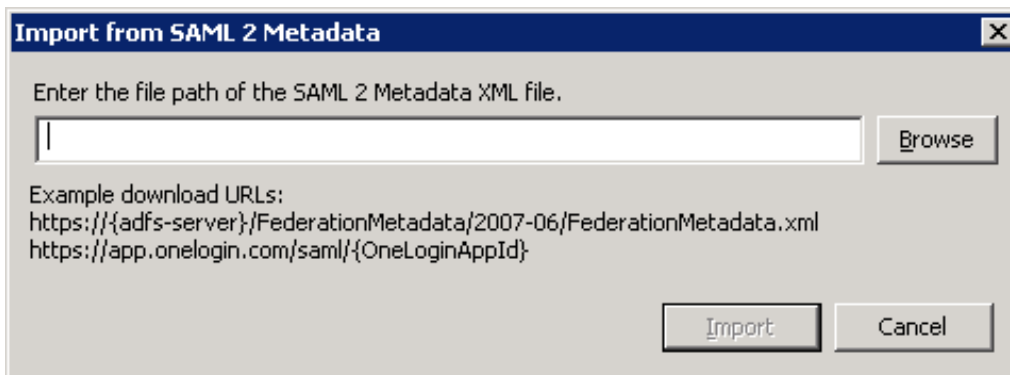
- In the left pane of the **Interaction Administrator** window, expand the **Single Sign-On** container and select the **Identity Providers** object.
- In the right pane of the **Interaction Administrator** window, right-click an empty area and select **New** from the resulting shortcut menu.

Interaction Administrator displays the **Identity Provider Configuration** dialog box.



- In the **Authentication Types** list, select the SAML profile and binding to use for this identity provider.
- Select the **Import** button.

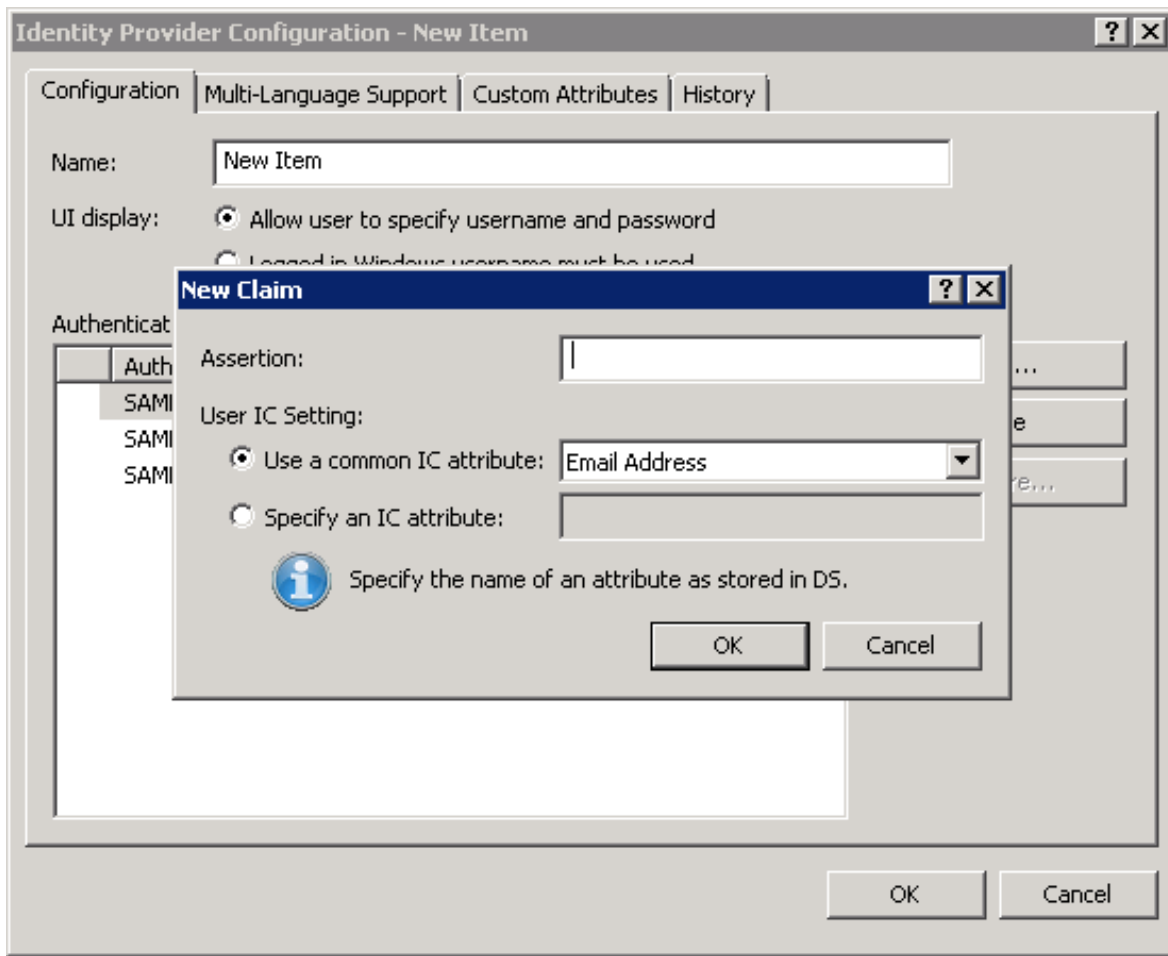
Interaction Administrator displays the **Import from SAML Metadata** dialog box.



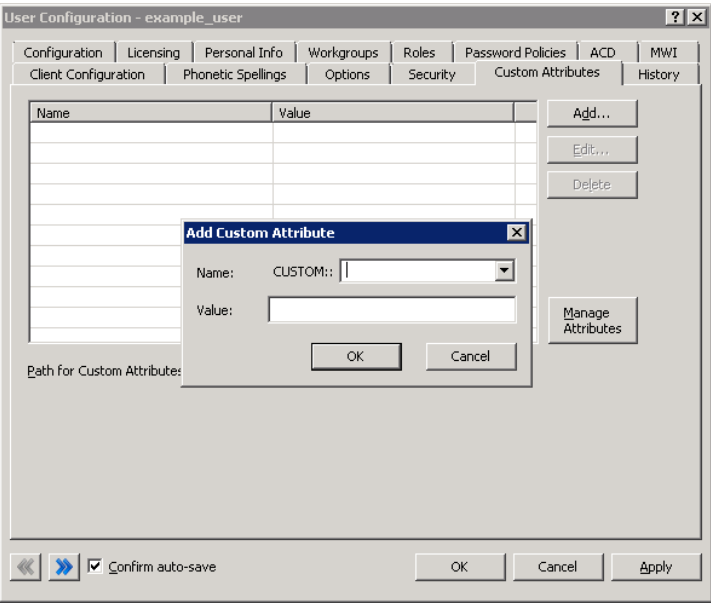
- In the available box, enter the location of the SAML 2 metadata file.
The location can be a local directory, UNC path for a network resource, or URL address.

Tip: You can use the **Browse** button to open a Windows Explorer window to manually navigate to the file.

- Select the **Import** button.
Interaction Administrator validates the contents of the SAML 2.0 metadata file and then displays the **New Claim** dialog box.



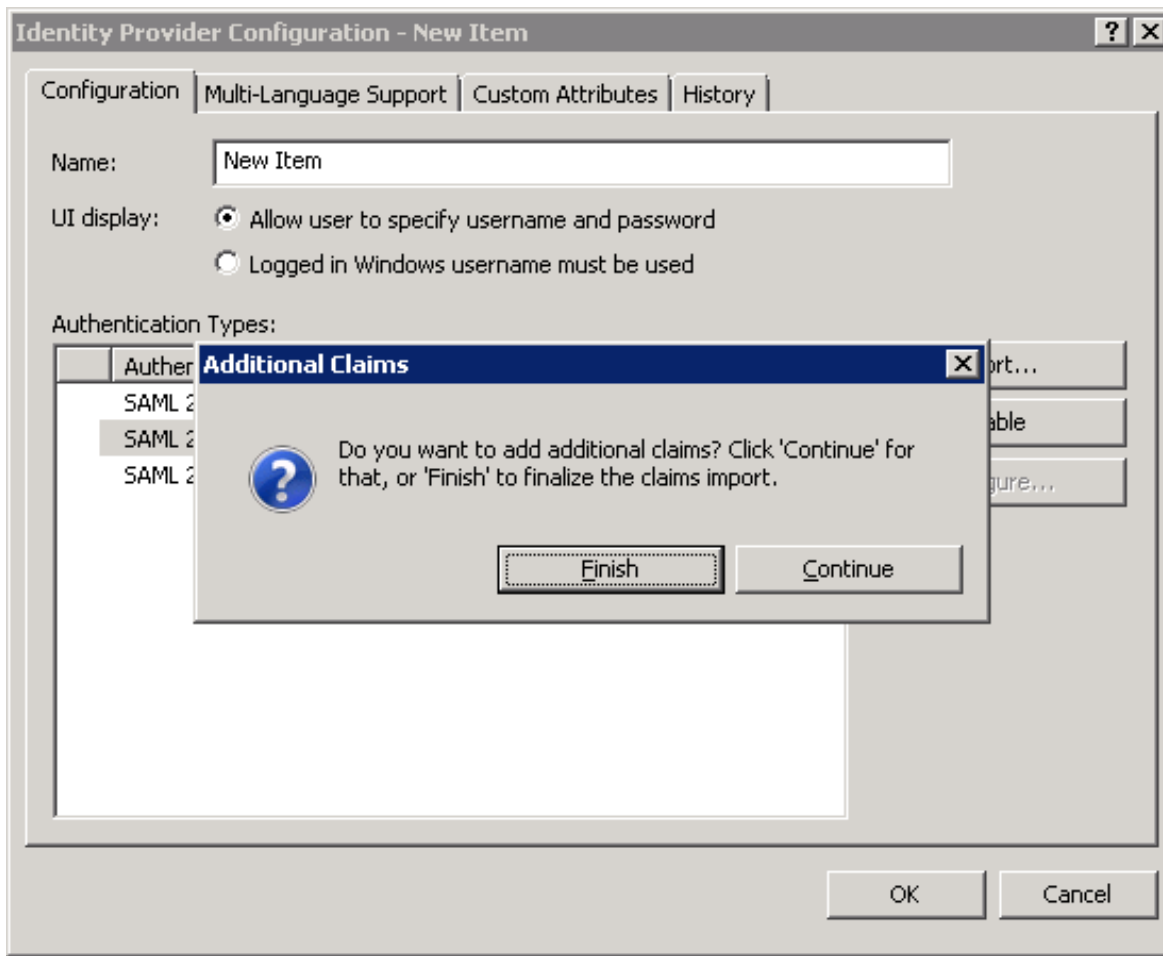
8. In the **Assertion** box, enter the Name attribute of an <Attribute> element that the identity provider will supply in SAML <AuthnRequest> responses.
9. In the **User IC Setting** group, select the option that enables you to map the SAML claim to the appropriate CIC attribute:

Option	Usage
Use a common IC attribute	<p>Select this option if you want to map the specified SAML claim to a CIC user attribute. Select that CIC user attribute in the list box:</p> <ul style="list-style-type: none"> • Email Address • User ID • Windows Domain Account <p>These attributes typically exist for each CIC user account.</p>
Specify an IC attribute	<p>Select this option if you want to map the specified SAML claim to a CIC Directory Services (DS) attribute that you previously defined.</p> <p>CIC Directory Services (DS) contain additional information on users, workgroups, workstations, lines, line groups, and other areas.</p> <p>For a partial list of CIC DS attributes for non-interaction objects, see "Attributes that can be looked up in Directory Services Keys" in <i>Interaction Designer Help</i>. You can also use <i>Interaction Designer Help</i> to find information on Interaction Designer tools that enable you to see the list of existing CIC Directory Services attributes.</p> <p>You can also assign SAML claims to custom CIC user attributes. To create a custom CIC user attribute, double click a user entry under People > Users in Interaction Administration, select the Custom Attributes tab, and then select the Add button.</p>  <p>When you specify this custom attribute in the Configuration dialog box for the selected SAML profile and binding, you must enter it in the following format:</p> <p>CUSTOM::<i>AttributeName</i></p> <p><i>AttributeName</i> is a variable that represents the name of the custom CIC user attribute that you defined.</p> <p>You must set the value of this customer attribute for each user entry under People > Users in Interaction Administrator.</p>

10. Select the OK button.

11. The claim is mapped to a CIC attribute and is saved as an entry in the list.

Interaction Administrator prompts you to determine if you need to enter additional claims from the identity provider.



12. If you need to enter another claim from the identity provider, select the **Continue** button and repeat steps 8 through 10. Otherwise, select the **Finish** button.

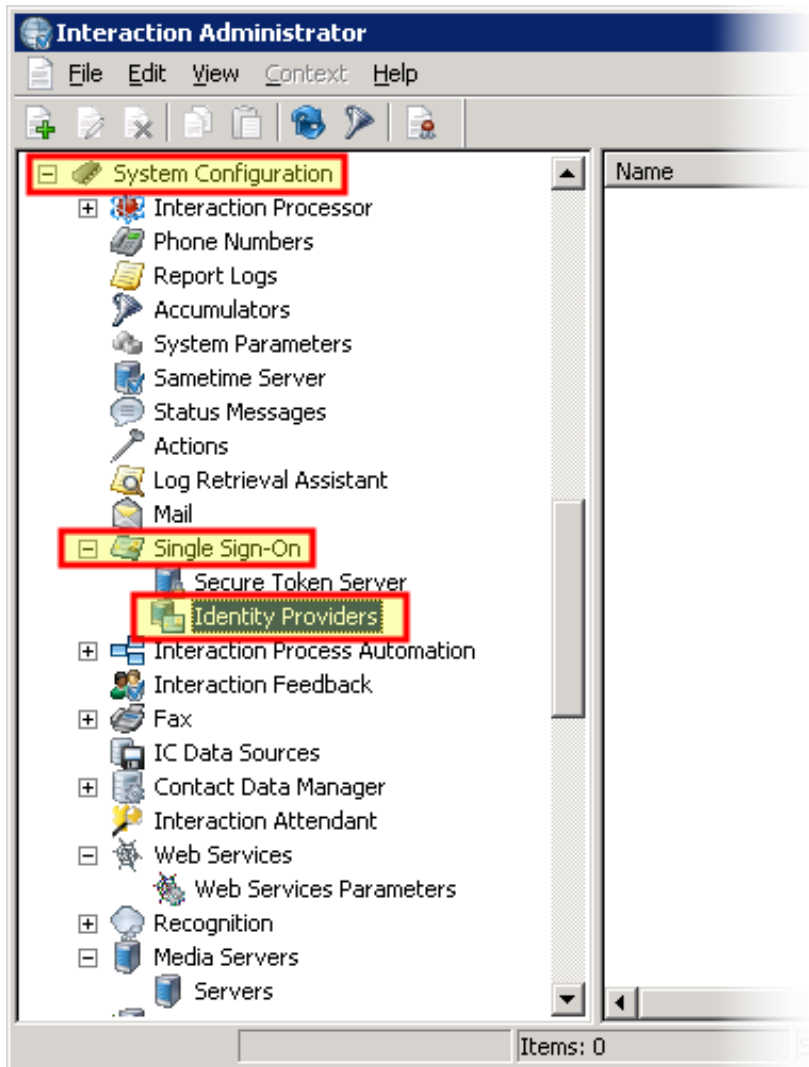
Interaction Administrator returns you to the **Identity Providers Configuration** dialog box.

13. Select the **OK** button to close the **Identity Providers Configuration** dialog box.

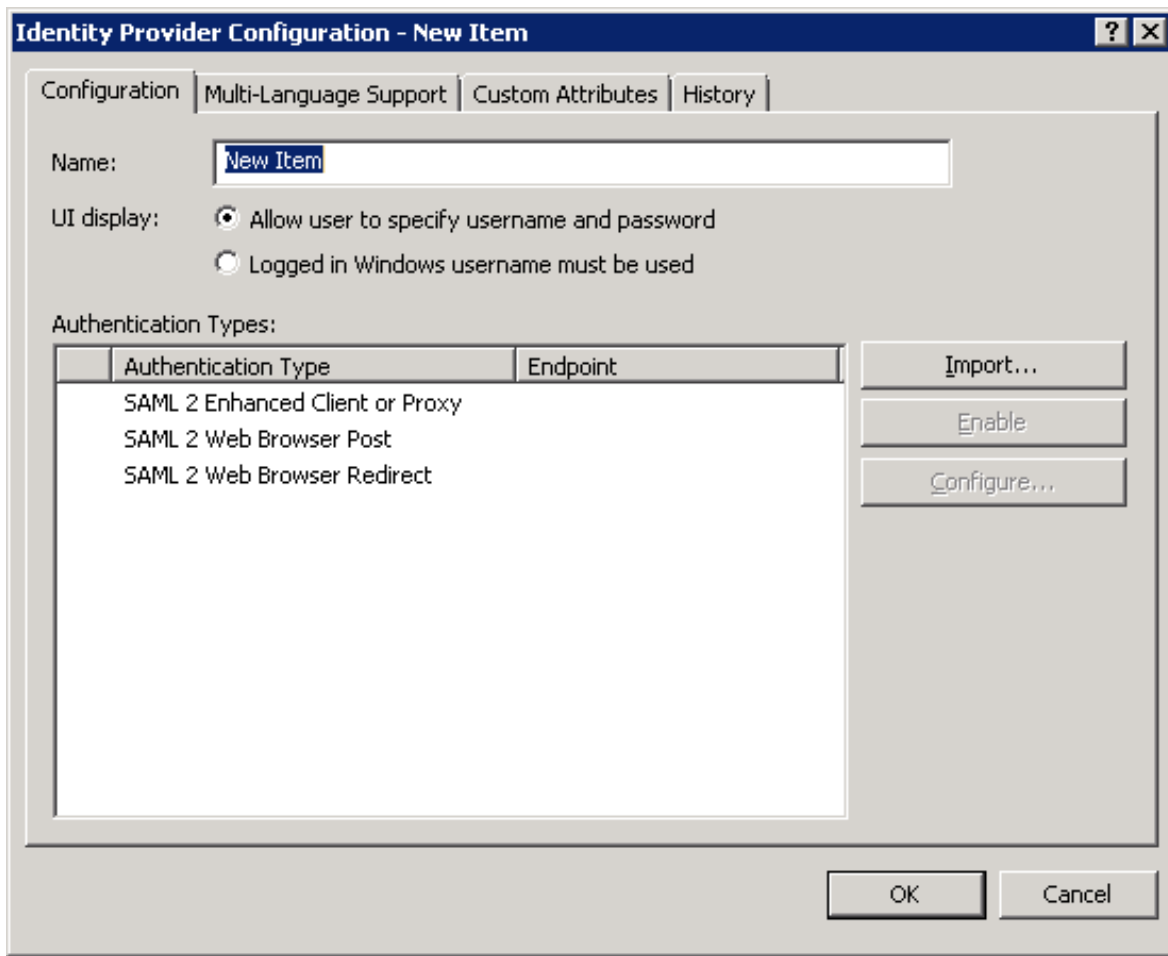
Manually configure identity provider settings

If the identity provider does not provide a SAML 2.0 metadata file, you must manually configure the CIC server through Interaction Administrator with the settings for that identity provider. This procedure requires the identity provider information that you obtained in [Gather identity provider information](#).

1. Open Interaction Administrator for the CIC server that will serve as the Single Sign-On service provider.
2. Log on to Interaction Administrator with an account that has administrative privileges.
3. In the navigation pane on the left side of the **Interaction Administrator** window, expand the **SystemConfiguration > Single Sign-On** object.
4. Under the **Single Sign-On** object, select the **Identity Providers** item.



5. In the right pane, right-click the empty area and select **New** from the resulting shortcut menu. Interaction Administrator displays the **Identity Provider Configuration** dialog box.



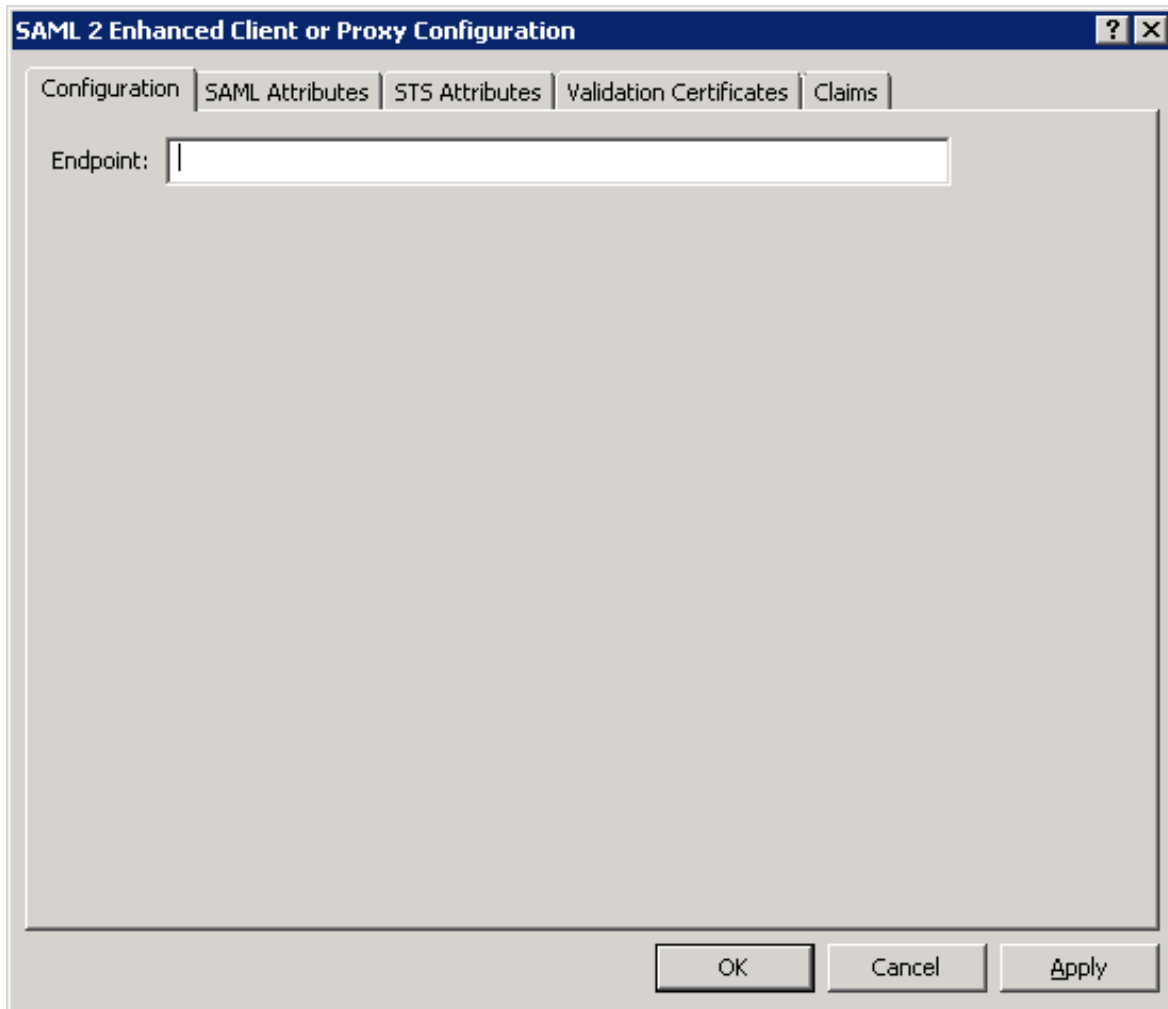
6. In the **Name** box, enter a name that you use for the identity provider.
The name in this box is only a label and does not need to match a character string from any file or gathered information.
7. Select the **Allow user to specify username and password** option.

In the **Authentication Types** list box, select a SAML binding and profile that you will use in your CIC environment and select the **Enable** button:

- **SAML 2 Enhanced Client or Proxy**
- **SAML 2 Web Browser Post**
- **SAML 2 Web Browser Redirect**

Note: For more information about these SAML bindings and profiles, see [Supported SAML bindings and profiles](#).

Interaction Administrator displays the **Configuration** dialog box for the SAML binding and profile that you selected.



Note: If you will use multiple SAML profiles and bindings in your CIC Single Sign-On environment, the identity provider may have a different address to which the client workstation sends SAML <AuthnRequest> messages and a separate validation certificate for each SAML profile and binding.

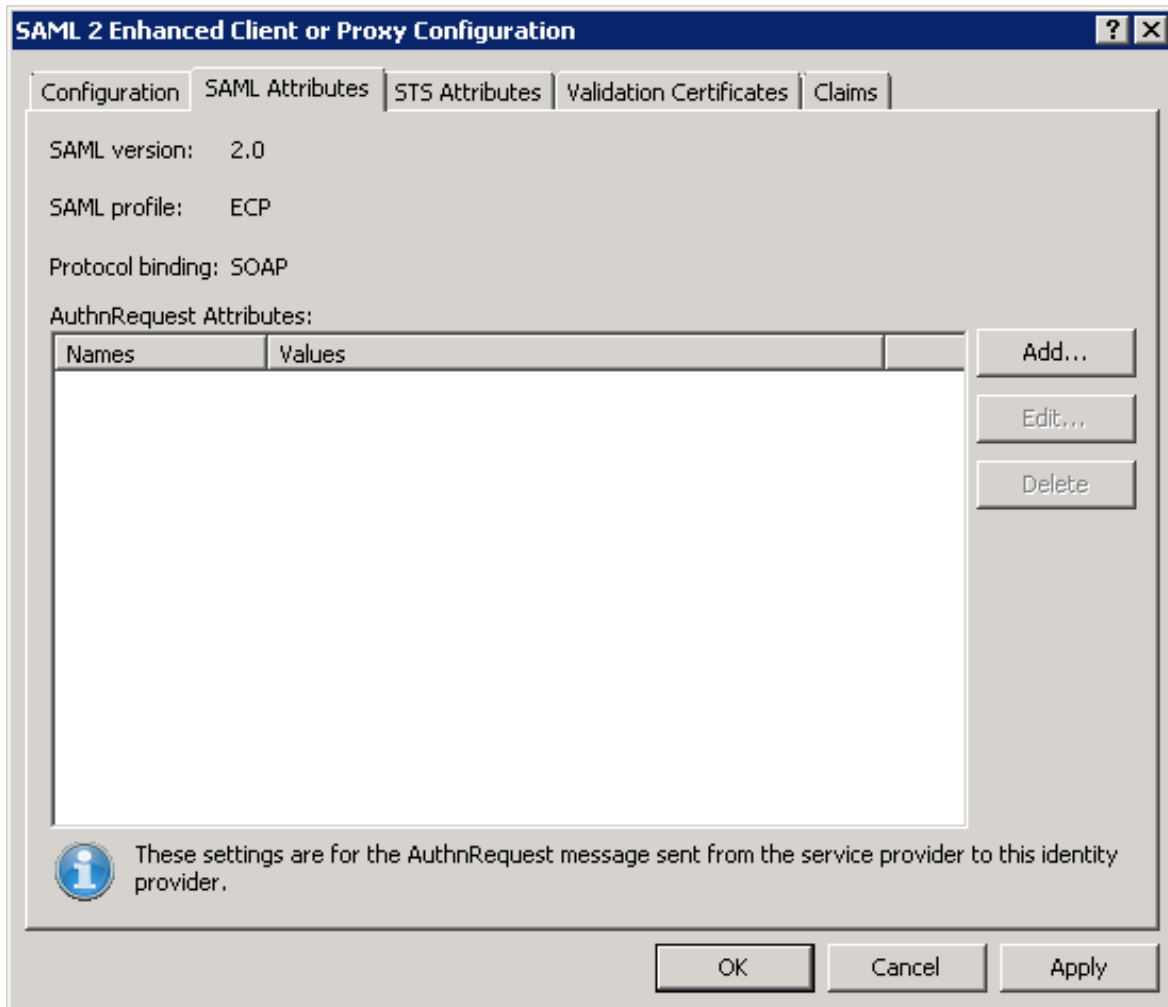
8. In the **Endpoint** box, enter the address—such as the URL address or FQDN—of the identity provider.

Note: Client workstations must be able to reach the address specified in the **Endpoint** box. Ensure that appropriate Domain Name Service settings are implemented.

9. Select the **Apply** button.
10. If your identity provider requires SAML authentication requests to be signed or requires specific attributes in SAML authentication requests, select the **SAML Attributes** tab and do the following procedure. Otherwise, continue to the next step.

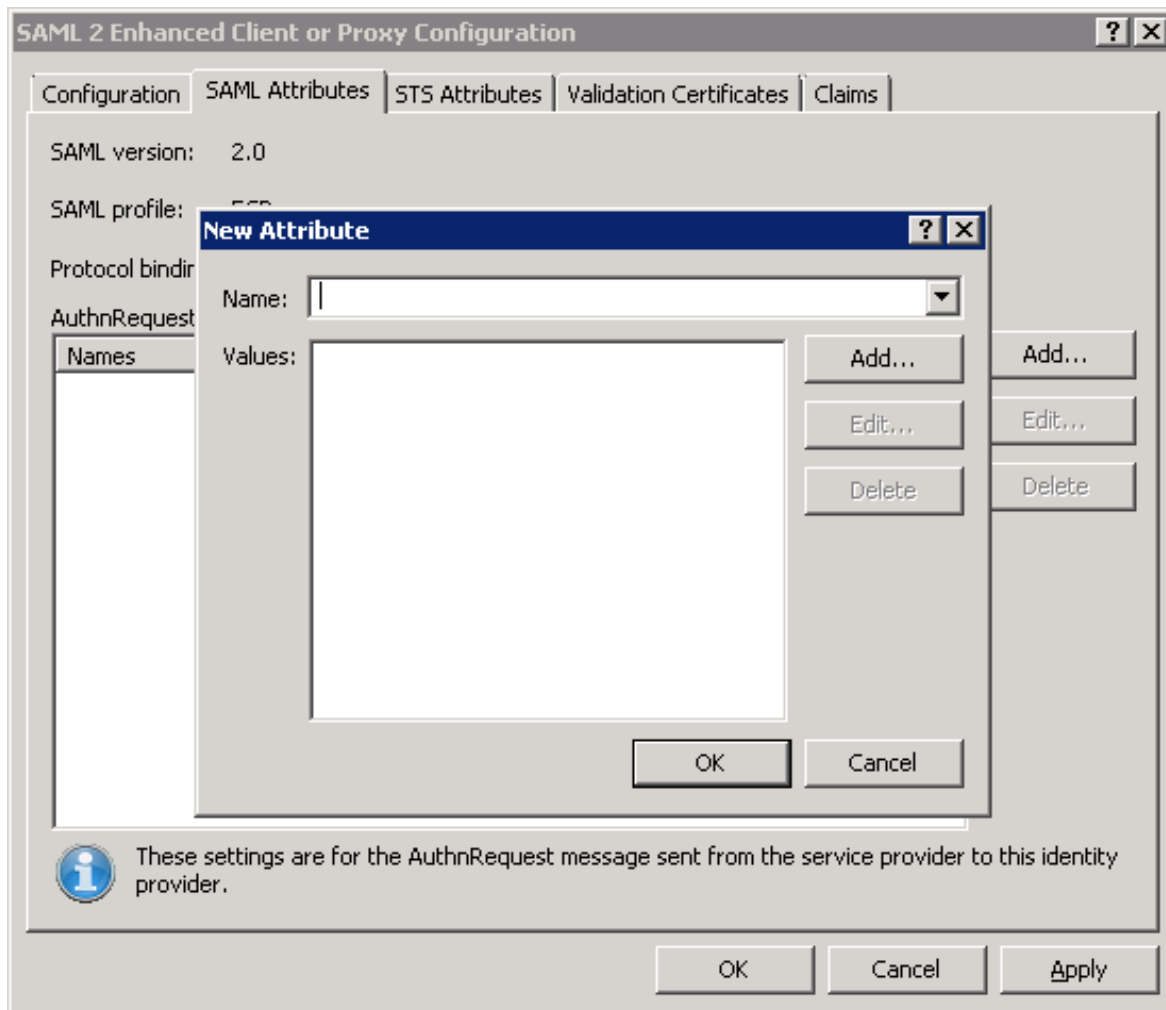
Note: Most identity providers do not require signed SAML <AuthnRequest> messages. Signing adds the CIC server signature in the message and requires the Trusted Certificate of the Certificate Authority that signed the following certificate: \\3\IC\Certificates\ICSecureTokenServer\Default\ICSecureTokenServerCertificate.cer

In most cases, the certificate is self-signed, so the Trusted Certificate for that certificate would be itself.



- a. Select the **Add** button.

The **New Attribute** dialog box is displayed.



- b. In the **Name** box, enter a required SAML attribute:

Important!

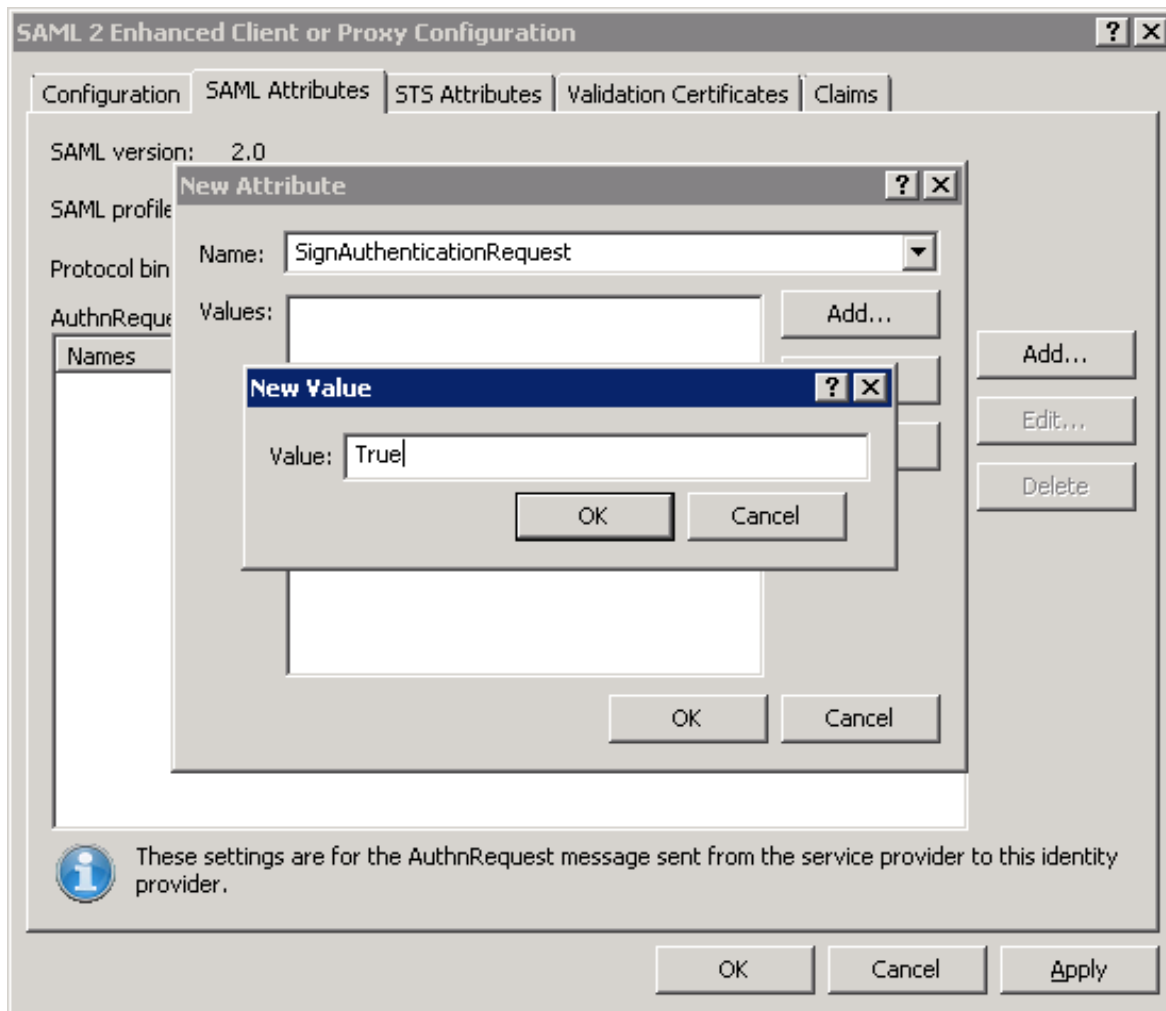
Improperly setting these SAML attributes can cause validation of your SAML <AuthnRequest> messages by your identity provider to fail. Genesys recommends that you access, read, and understand the following specification before attempting to set SAML attributes:

<https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

SAML attribute	Description
ID	By default, the CIC server sets the value of this SAML attribute to a unique identifier but you can configure it to contain a specific value.
Version	By default, the CIC server sets the value of this SAML attribute to "2.0" but you can configure it to contain a specific value.
Destination	By default, the CIC server sets the value of this SAML attribute to the endpoint of the identity provider but you can configure to contain a different value.
Consent	By default, the CIC server sets this SAML attribute to the following value: urn:oasis:name:tc:SAML:2.0:consent:unspecified You can configure this attribute to contain a different value.
ForceAuthn	By default, the CIC server sets the value of this SAML attribute to false but you can configure it to contain a different value.
IsPassive	By default, the CIC server sets the value of this SAML attribute to false but you can configure it to contain a different value.
AssertionConsumerServiceIndex	By default, the CIC server does not use this SAML attribute but you can specify a single value as directed by your identity provider. Note: If you specify this SAML attribute, the CIC server will not include the AssertionConsumerServiceURL and ProtocolBinding attributes.
AssertionConsumerServiceURL	By default, the CIC server sets the value for this SAML attribute to the address you determined in Assertion Consumer Service URL . However, you can specify a different value if required by your identity provider.
ProtocolBinding	By default, the CIC server sets this SAML attribute to the following value: urn:oasis:names:tc:SAML:2.0:bindings:AuthenticationType <i>AuthenticationType</i> is a variable representing the SAML profile and binding through which the SAML <AuthnRequest> message is sent.
AttributeConsumingServiceIndex	By default, the CIC server does not include this SAML attribute but you can set it with a specific value, which the CIC server will include in SAML <AuthnRequest> messages to the identity provider.
ProviderName	By default, the CIC server sets this SAML attribute to the following value: <code>https://CICServerEndpoint.Port/AuthenticationType/login</code> Should your identity provider require it, you can override this value.
NameIDPolicy	By default, the CIC server does not use this SAML attribute but you can specify a value that the CIC server will send in all SAML <AuthnRequest> messages to the identity provider.
RemoveIssueInstantMilliseconds	By default, the CIC server sets this value of this attribute to false. You can set the value of this attribute to always be true should your identity provider require it. Note: This attribute is an add-on attribute from Genesys and is not a standard SAML attribute.
SignAuthenticationRequest	By default, the CIC server sets the value of this attribute to false but, if your identity provider requires signed SAML <AuthnRequest> messages, you can set this value to true. Note: This attribute is an add-on attribute from Genesys and is not a standard SAML attribute.

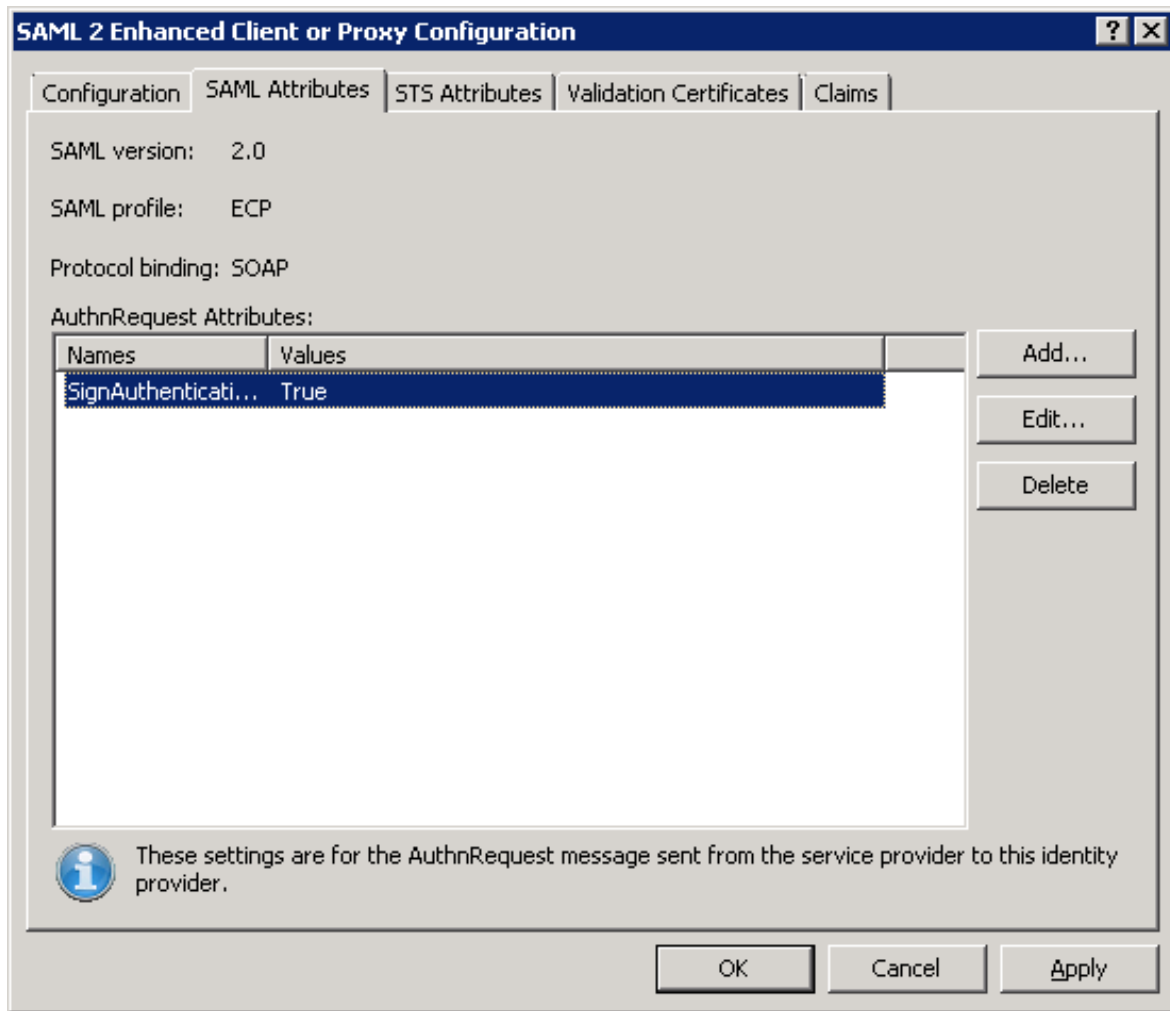
c. Select the Add button.

The **New Value** dialog box is displayed.



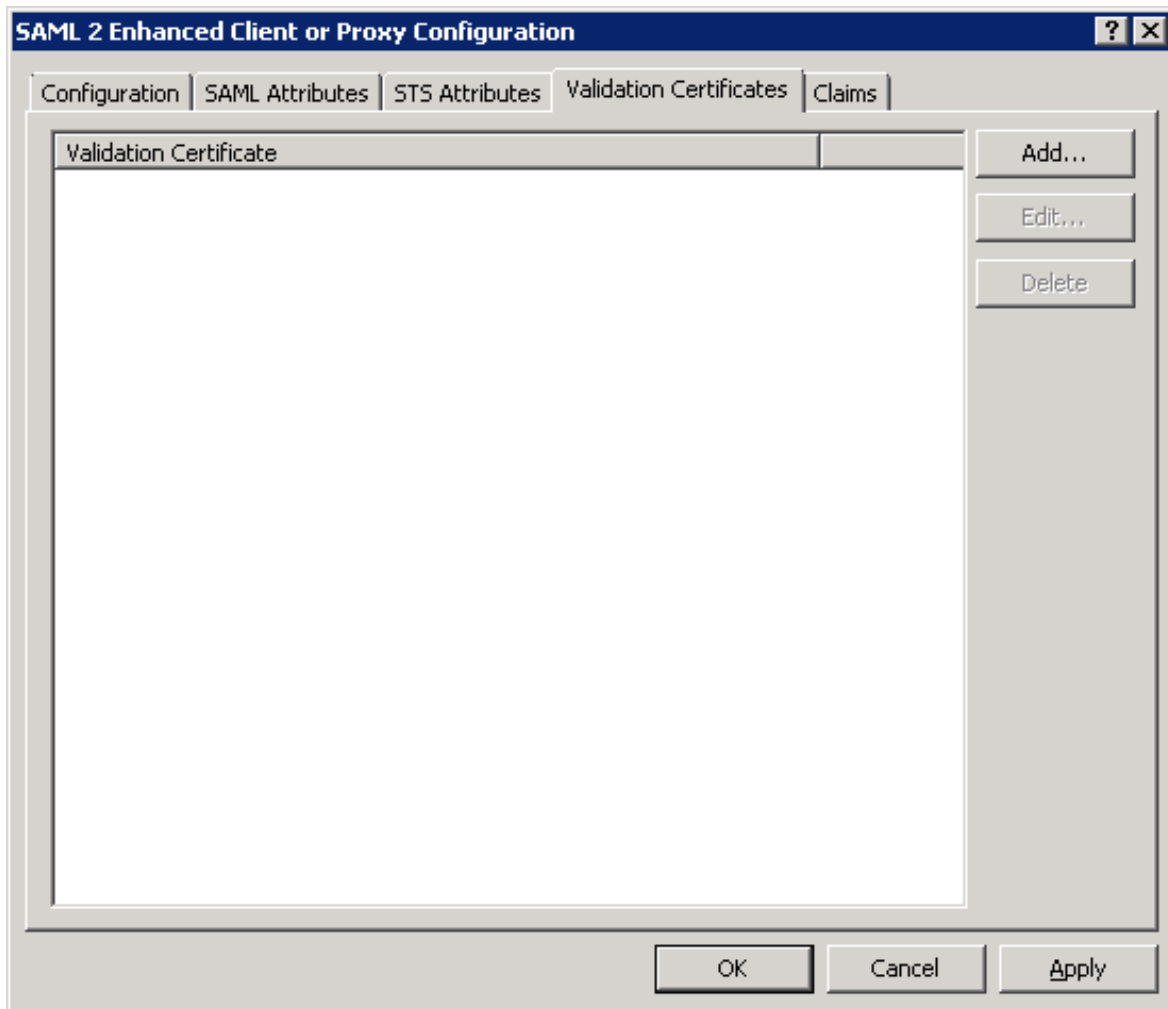
- d. In the **Value** box, enter the required value for this SAML attribute.
- e. In the **New Value** dialog box, select the **OK** button.
- f. In the **New Attribute** dialog box, select the **OK** button.

The attribute and value are added to the **AuthnRequest Attributes** list box.



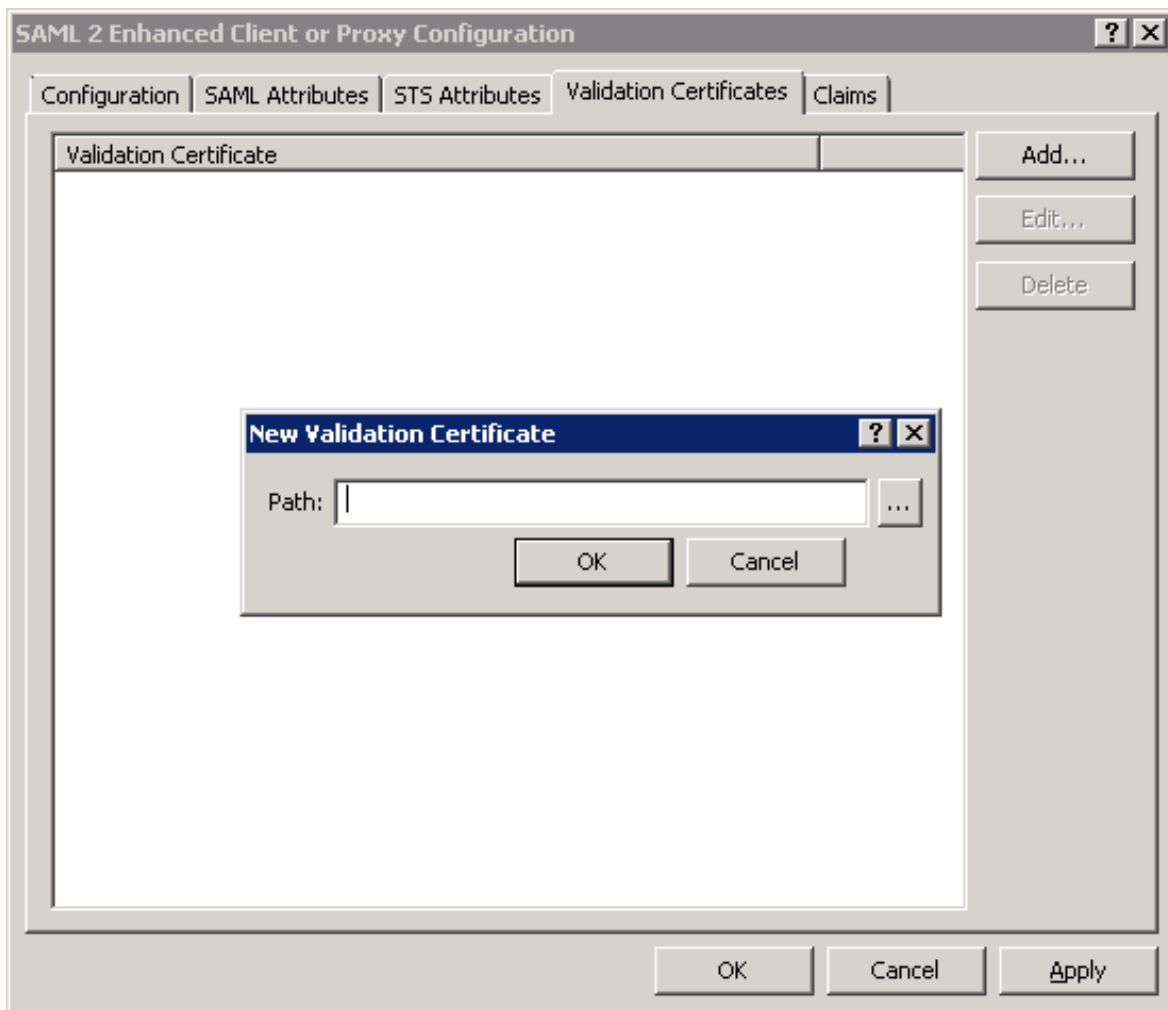
g. Repeat steps a through e for any additional SAML attributes that are required.

11. Select the **Validation Certificates** tab.



12. Select the **Add** button.

Interaction Administrator displays the **New Validation Certificate** dialog box.



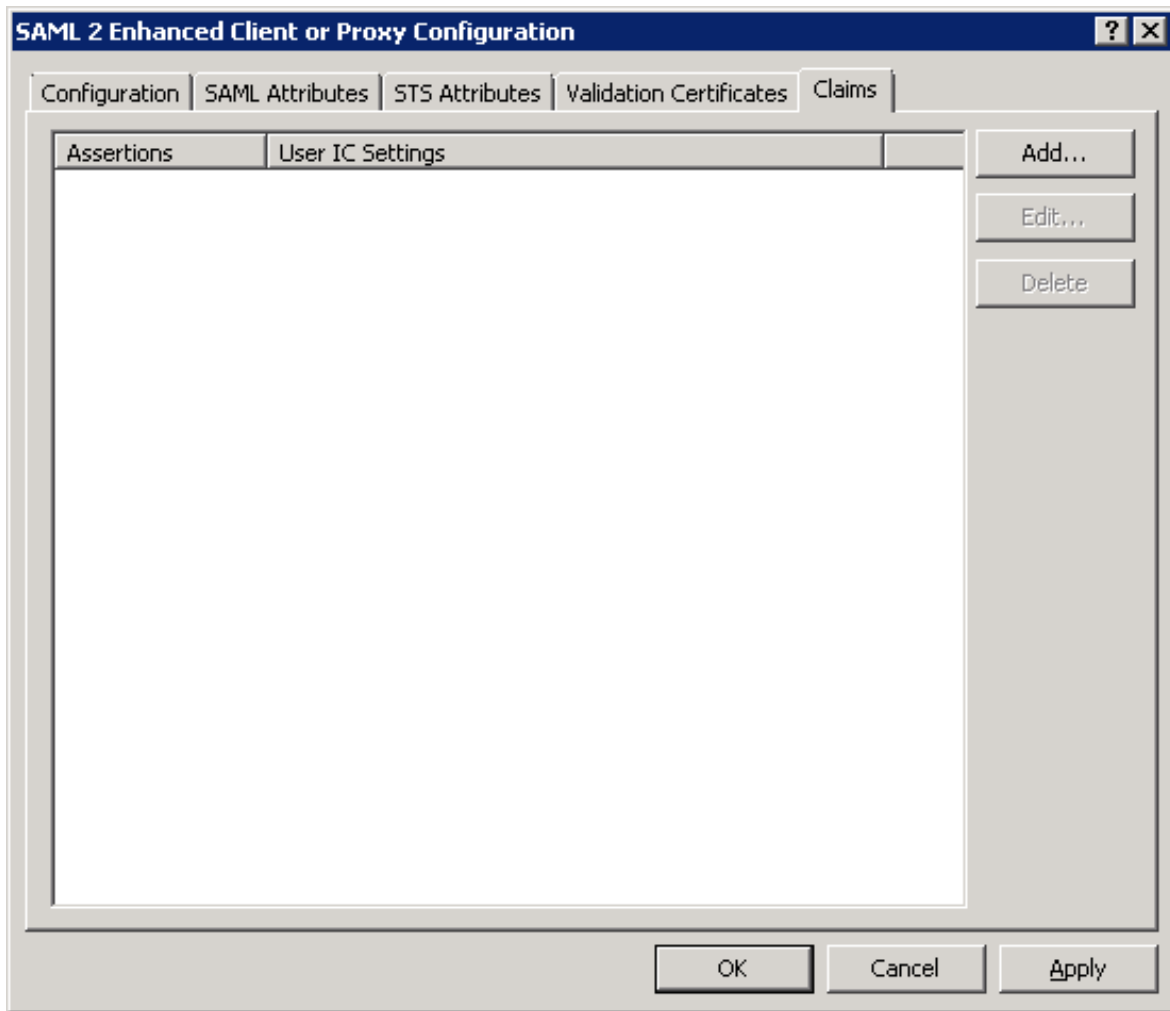
13. In the **Path** box, enter the location of the validation certificate that you received from the identity provider.

Tip: You can select the button to the right of the **Path** box to browse for the file.

Important!

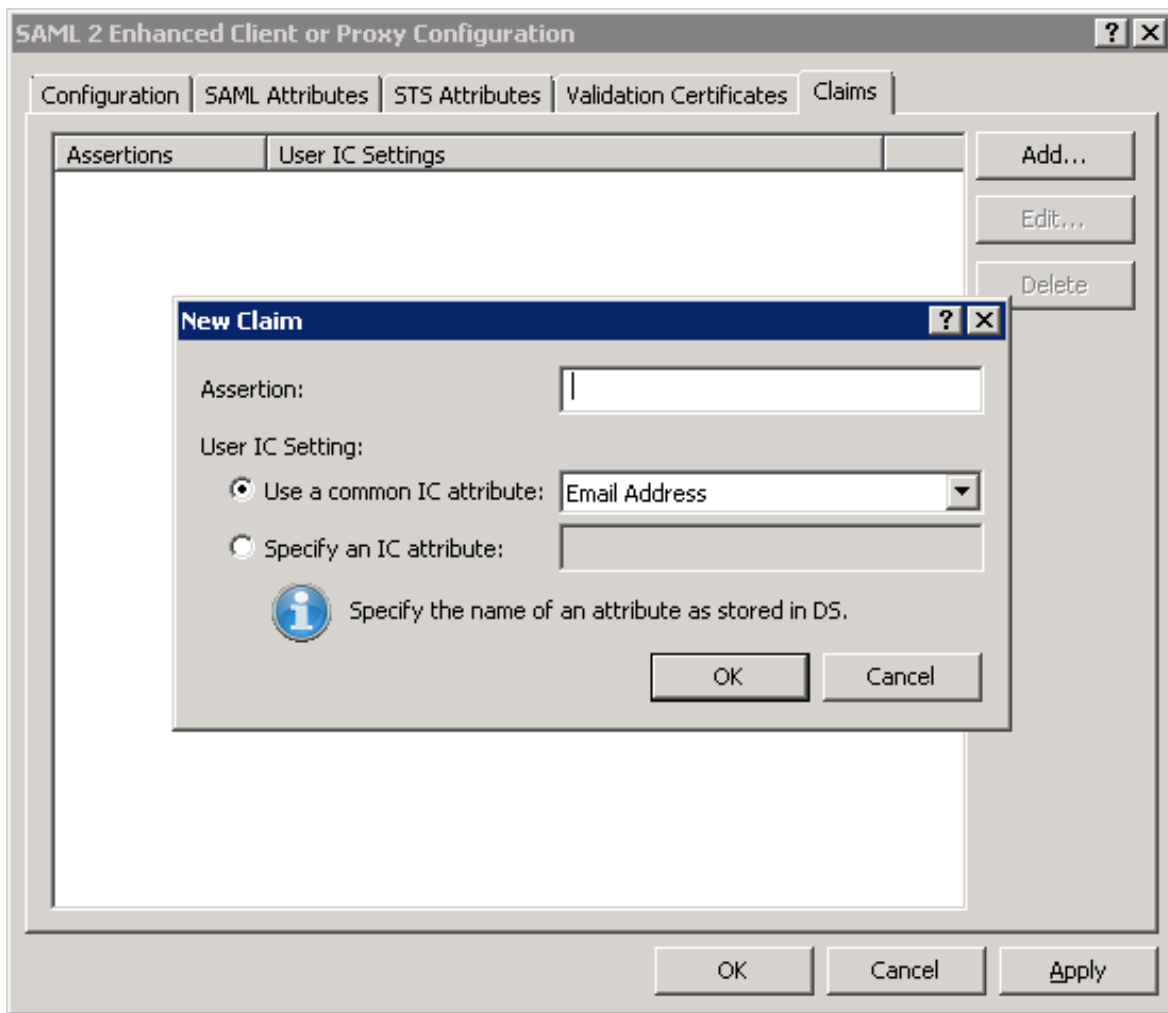
The validation certificate from the identity provider must be a valid X.509 certificate or certificate chain and must be in PEM (base-64) format. To validate the format of the validation certificate, see [Ensure the format of the validation certificates](#).

14. Select the **OK** button.
15. Select the **Claims** tab.



16. Select the **Add** button.

The **New Claim** dialog box is displayed.



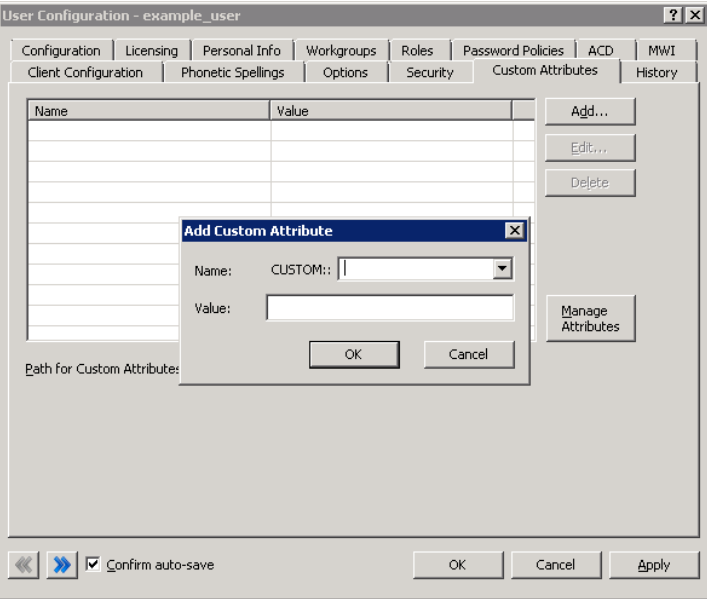
17. In the **Assertion** box, enter the Name attribute of an <Attribute> element that the identity provider will supply in SAML <AuthnRequest> responses.

Example:

The following SAML claim is common as it represents a Windows user account name:

<http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname>

18. In the **User IC Setting** group, select the option that enables you to map the SAML claim to the appropriate CIC attribute:

Option	Usage
Use a common IC attribute	<p>Select this option if you want to map the specified SAML claim to a CIC user attribute. Select that CIC user attribute in the list box:</p> <ul style="list-style-type: none"> • Email Address • User ID • Windows Domain Account <p>These attributes typically exist for each CIC user account.</p>
Specify an IC attribute	<p>Select this option if you want to map the specified SAML claim to a CIC Directory Services (DS) attribute that you previously defined.</p> <p>CIC Directory Services (DS) contain additional information on users, workgroups, workstations, lines, line groups, and other areas.</p> <p>For a partial list of CIC DS attributes for non-interaction objects, see "Attributes that can be looked up in Directory Services Keys" in <i>Interaction Designer Help</i>. You can also use <i>Interaction Designer Help</i> to find information on Interaction Designer tools that enable you to see the list of existing CIC Directory Services attributes.</p> <p>You can also assign SAML claims to custom CIC user attributes. To create a custom CIC user attribute, double click a user entry under People > Users in Interaction Administration, select the Custom Attributes tab, and then select the Add button.</p>  <p>When you specify this custom attribute in the Configuration dialog box for the selected SAML profile and binding, you must enter it in the following format:</p> <p>CUSTOM::<i>AttributeName</i></p> <p><i>AttributeName</i> is a variable that represents the name of the custom CIC user attribute that you defined.</p> <p>You must set the value of this customer attribute for each user entry under People > Users in Interaction Administrator.</p>

19. Select the **OK** button.

The claim is mapped to a CIC attribute and is saved as an entry in the list.

20. Repeat steps 16 through 19 for each additional claim that you want to configure.

21. When you have finished adding claims, select the **OK** button.

Provide CIC Single Sign-On information to identity provider

Your identity provider requires the following items that you collected in [Gather CIC server endpoint information](#) and created in [Configure Single Sign-On for a CIC system](#):

Item	Description
Assertion Consumer Service URL	<p>The ACS URL is a combination of a URL address, port number, SAML profile and binding identifier, and a directory specific to CIC Single Sign-On.</p> <p>Examples:</p> <ul style="list-style-type: none"> https://cic.example.com:8043/SAML2WebBrowserPostHTTPS/login https://icws.example.com:8019/icws/connection/single-sign-on/return https://connect.example.com/api/icws.example.com/icws/connection/single-sign-on/return <p>For more information about determining your ACS URL, see Assertion Consumer Service URL and ACS URL example configurations.</p>
Issuer	The identity provider may use any one of the names in the Item column for this item.
Provider Name	This item is a simple address, which is the combination of the CIC server address that you determined in Single Sign-On configurations and the port number you determined in Determine default port for Single Sign-On..Supply this item in the following format:
Relaying Party Identifier	https://CICServerAddress:Port
Partner Identifier	Example:
Entity ID	https://cic.example.com:8043
HTTPS certificate for the CIC server	You created the certificate in Generate a self-signed HTTPS certificate .Ensure that you copy the certificate from the correct directory on the CIC server that you used in the referenced procedure.
CIC Server Trusted Certificate	<div style="border: 2px solid red; padding: 5px;"> <p>Important!</p> <p>This item is required only for those identity providers that require SAML <AuthnRequest> messages be signed.</p> </div> <p>This item is the \I3\IC\Certificates\ICSecureTokenServer\Default\ICSecureTokenServerCertificate.cer certificate.</p>

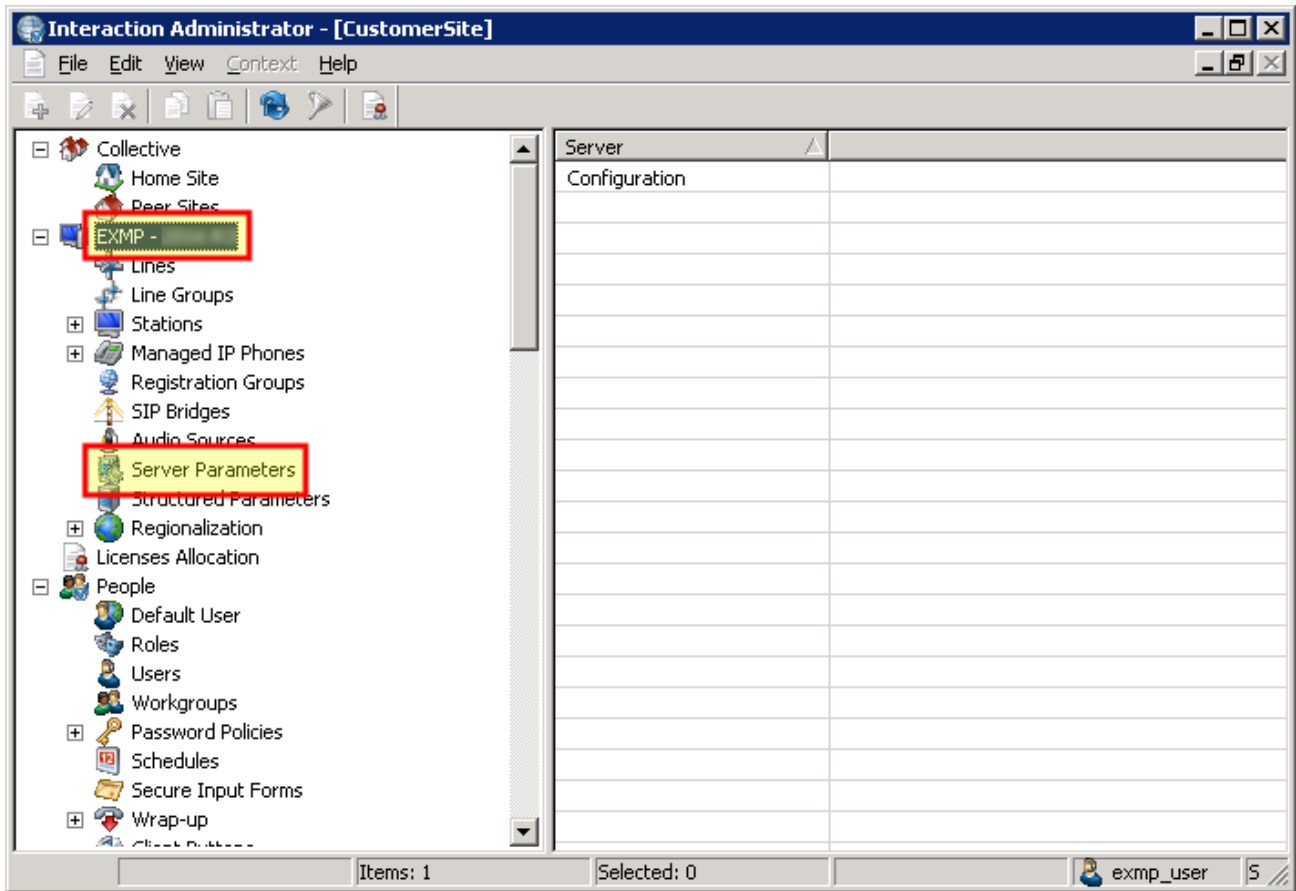
Single Sign-On Configuration Utility

The Single Sign-On Configuration Utility does the following actions:

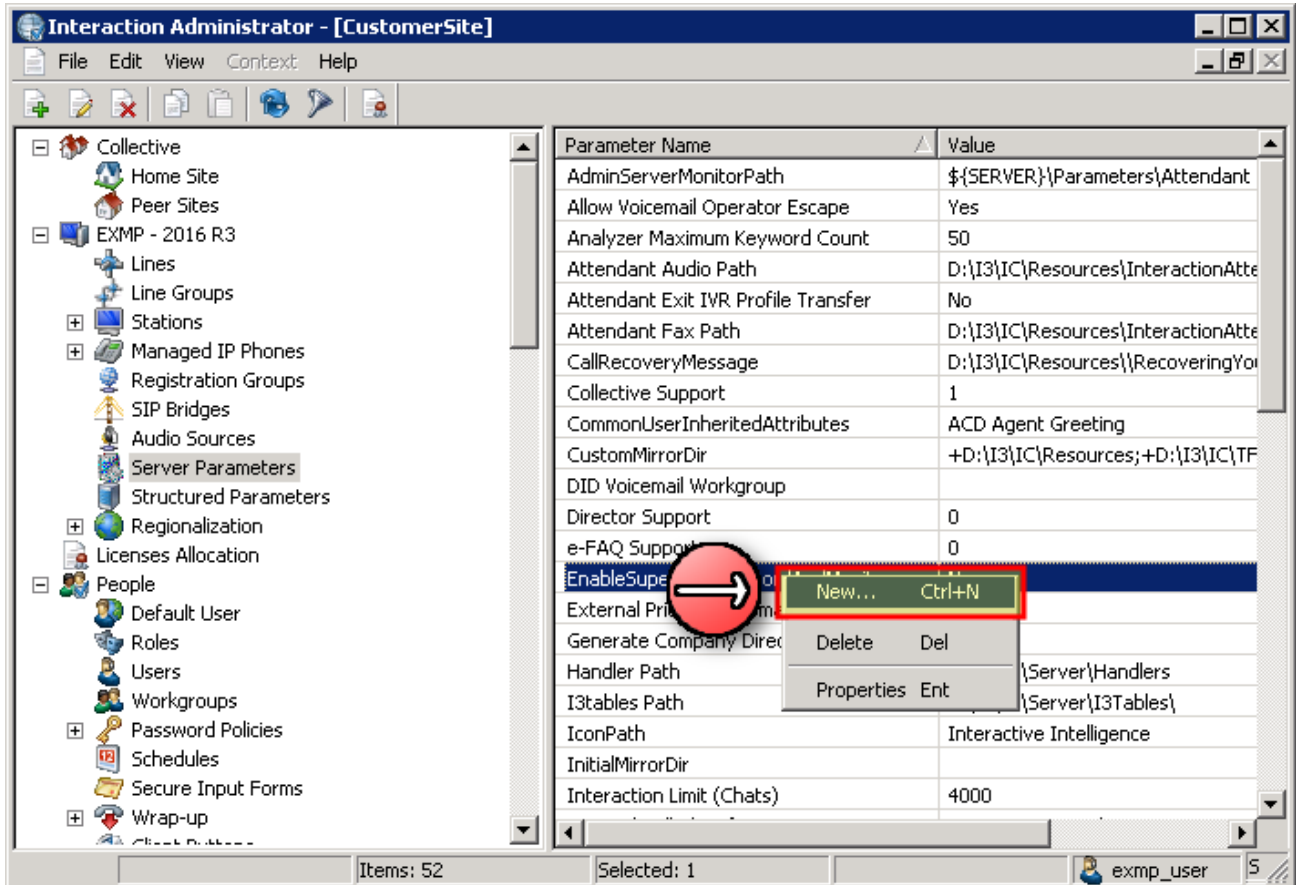
- Creates SAML metadata for Single Sign-On identity providers
- Downloads an HTTPS certificate that you can apply to computers with CIC client applications or proxies to enable Single Sign-On logon capabilities.

Enable SSO Configuration Utility through Interaction Administrator

- Open Interaction Administrator and log on using administrative credentials.
The Interaction Administrator window appears.
- In the navigation pane on the left side of the Interaction Administrator window, expand the container that represents your CIC server.
- Under the CIC server container, select the **Server Parameters** container.

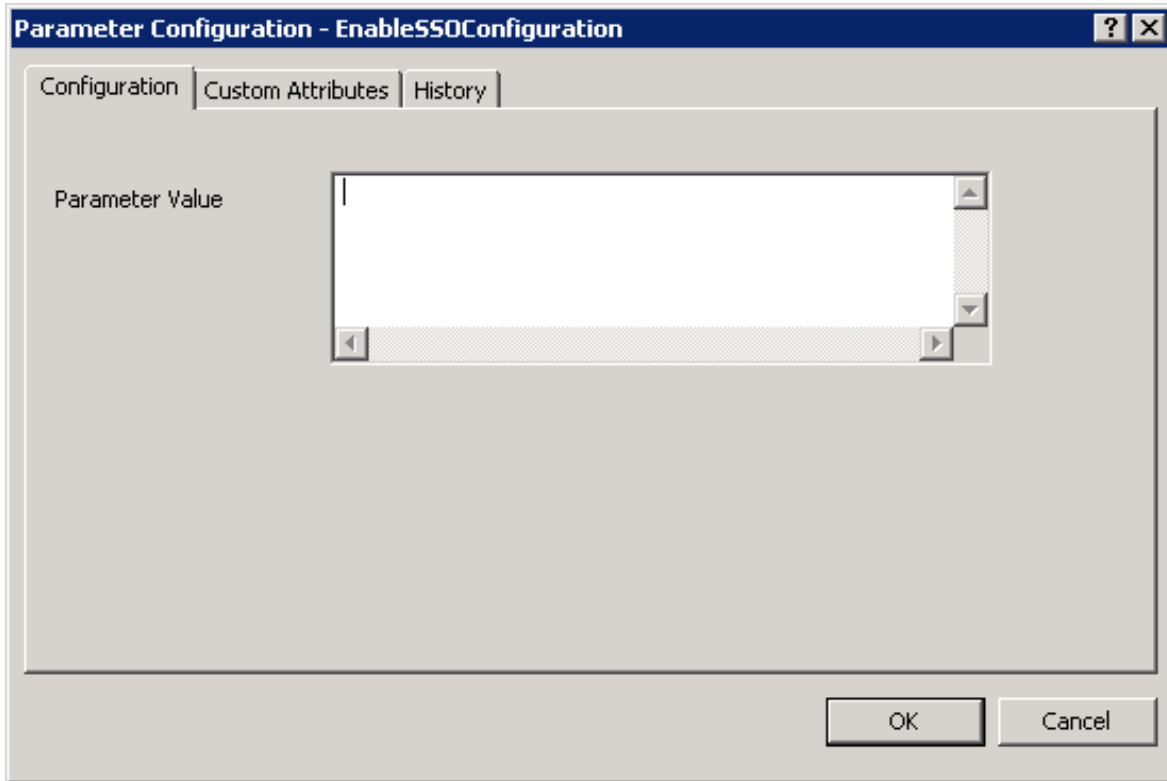


4. In the right-pane, right-click and select New from the resulting context menu.



Interaction Administrator displays an Entry Name dialog box.

5. In the **Entry Name** dialog box, enter **EnableSSOConfiguration** and select **OK**.
Interaction Administrator displays the **Parameter Configuration** dialog box.



6. In the **Parameter Value** box of the **Parameter Configuration** dialog box, enter **True** and select **OK**.
Interaction Administrator creates the **EnableSSOConfiguration** server parameter and displays it.

Use SSO Configuration Utility on a client workstation

This procedure uses SSO Configuration Utility to configure the appropriate information so that users can authenticate against the CIC server using Single Sign-On.

1. On a client workstation within the network where CIC resides, open a web browser and navigate to the following URL address:

`https://<CIC server IP address>:8043/SSOConfiguration/saml`

<CIC server IP address> is the IP address of the CIC server.

Note: Network port 8043 must not be blocked by a firewall on the workstation or a network entity between the workstation and the CIC server.

The web browser displays the first page of the SSO Configuration Utility.

SAML Single Sign-On Configuration.

This utility will configure the SAML based Single Sign-On mechanism for your Interaction Center Server.

Please run this utility on a Client machine, or on a machine in the same domain as the Client machine.

Next

2. Select **Next**.

The first question is displayed.

1.) Is this a stand-alone Interaction Center Server?

- Yes, it is a stand-alone Interaction Center Server.
- No, this Interaction Center Server is part of a SwitchOver pair.

The IC Secure Token Server on the Interaction Center Server is listening on port:

Next Back Close

3. Select the option that reflects your CIC server configuration:

- **Yes** The CIC server is not part of a switchover pair.

If you select the **Yes** option, the page displays a box for you to enter the name of the CIC server.

- Yes, it is a stand-alone Interaction Center Server.

The Interaction Center Server is:

- No The CIC is part of a switchover pair

If you select the No option, the page displays a box for you to enter the DNS A/AAAA-record for the switchover pair.

- No, this Interaction Center Server is part of a SwitchOver pair.

The DNS-A/AAAA record for the SwitchOver pair is:

(example: caasprovision.caas.local)

- In the available box, enter the port number that the IC Secure Token Server uses for Single Sign-On (SAML) communications. The default port is 8043.

- Select **Next**.

The test page is displayed.

2.) Testing connection to the Interaction Center Server.

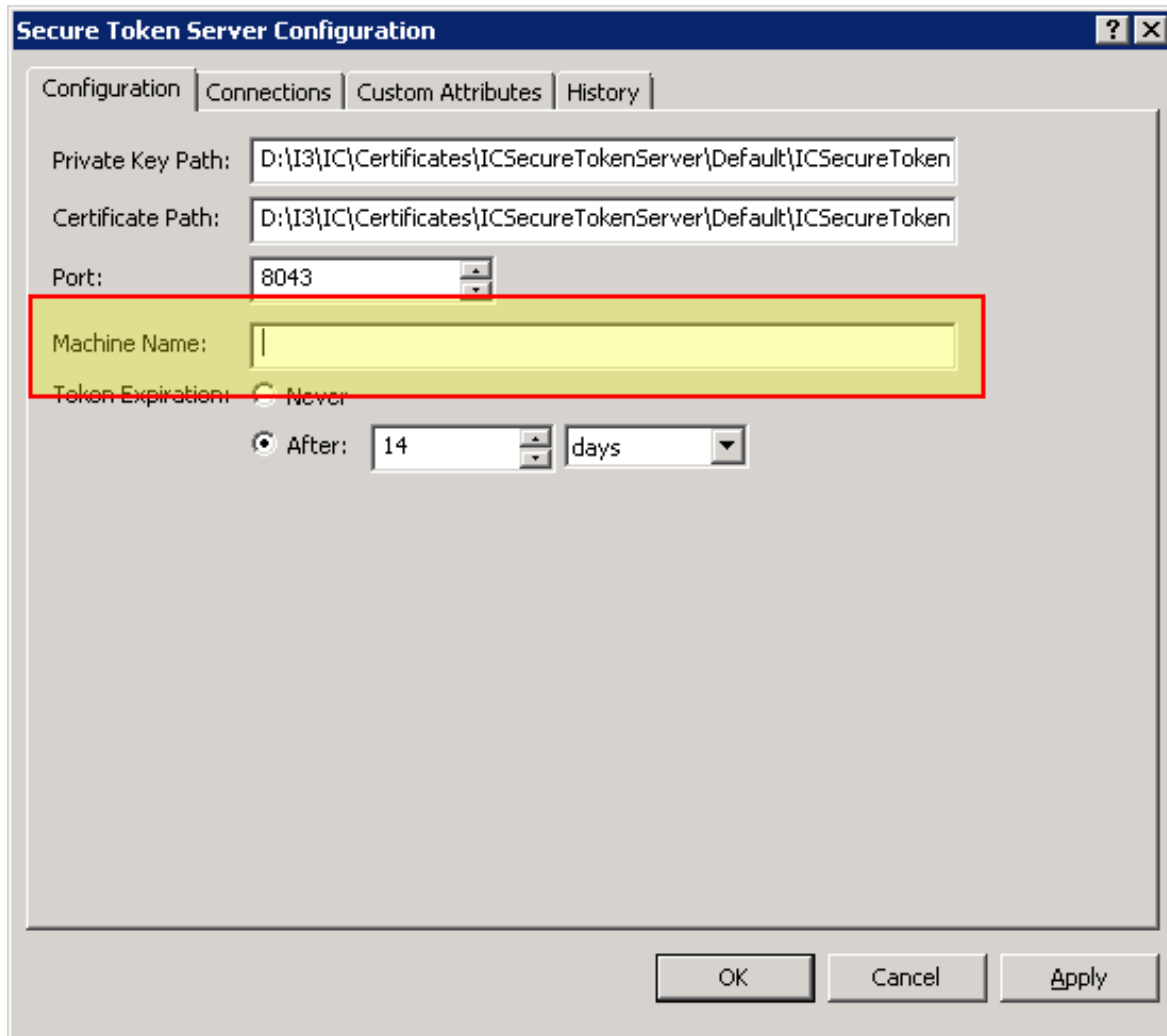
Connect to:

You must test the connectivity to the CIC server before the **Next** button will be enabled.

- Verify that the address and port number in the **Connect to** box are correct and select **Test**.

If the test was unsuccessful, the problem is usually because of one the following conditions:

- The address of the CIC server (IP address or Fully-Qualified Domain Name) does not match the entry in the **Machine Name** box of the **Secure Token Server Configuration** dialog box in Interaction Administrator. You entered the **Machine Name** in the [Configure Secure Token Server](#) procedure.



- The name of the CIC server does not match the one in the [HTTPS certificate for the CIC service provider](#) that you created in the [Administer HTTPS certificate for the CIC service provider](#) section of this document. For this result, select the **Download HTTPS Certificate Generating Batch File** button. Move the downloaded file to the CIC server and run it on the CIC server.

2.) Testing connection to the Interaction Center Server.

Connect to:

Test

The Interaction Center Server name that you have specified (i.e: 10.8.3.154) is not specified as either the 'Common Name' or as a 'Subject Alternative Name' in the HTTPS certificate: *D:\I3\IC\Certificates\HTTPS\CLAY_Certificate.cer*. To correct this, please download the following batch file, and **run it on the Interaction Center Server** before continuing.

Download HTTPS Certificate Generating Batch File

Next

Back

Close

7. If the test was successful select **Next**.
The next page of SSO Configuration Utility is displayed.

3.) Configure proxied connections to the Interaction Center Server.

Add any reverse proxy connections to the Interaction Center Server (e.g: *https://cl-ubuntuws1.ininlab.com/webic_client_systest/latest/client/api/Jenova*, *https://ossm3.dev2000.com/JENOVA.DEV2000.COM/httpsecure*, *https://myreverseproxy/api/myICServerName* etc ...). Click 'Next' if there aren't any.

Next

Back

Close

8. If you use any reverse proxies in your network, enter their URL addresses in the box provided.
9. Select **Next**.
The final page of SSO Configuration Utility is displayed.

4.) Download and apply the SAML configuration.

Please download the SAML metadata file and apply it on your Identity Provider.

Download SAML Metadata

Please download the HTTPS certificate and apply it on **all of your Client machines and proxies**.

Download HTTPS Certificate

Once the SAML configuration has been applied on your Identity Provider and Client machines, you may download and run the Single Sign-On Testing Utility below to verify the Identity Providers configuration on your Interaction Center Server.

Download Single Sign-On Testing Utility (for Windows)

Next

Back

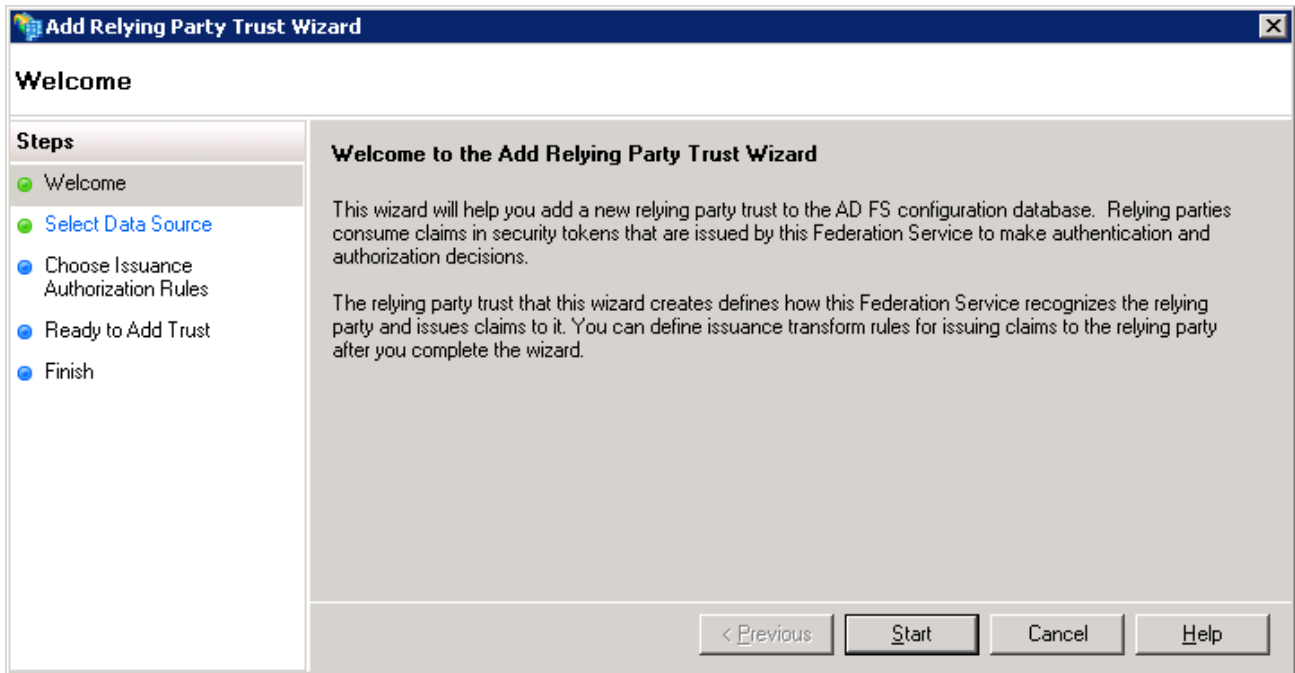
Close

10. Select the **Download SAML Metadata** button and apply the downloaded file on your identity provider. If you are using a third-party identity provider, provide this file to the provider.
11. Select the **Download HTTPS Certificate** button and apply it to all computers with CIC clients or proxies using one of the procedures in [Import the HTTPS certificate of the CIC server onto workstations hosting CIC client applications](#).
12. Optionally, after you apply the SAML Metadata to your identity provider, you can download and run the Single Sign-On Testing Utility on Windows-based computers by selecting the **Download Single Sign-On Testing Utility (for Windows)** button. For more information about Single Sign-On Testing Utility (for Windows), see [Test Single Sign-On for the identity provider](#).

Configure Microsoft AD FS as an identity provider

1. Open the Microsoft AD FS Management snap-in.
2. In the left pane of the AD FS Management snap-in window, right-click and select **Add Relying Party Trust**.

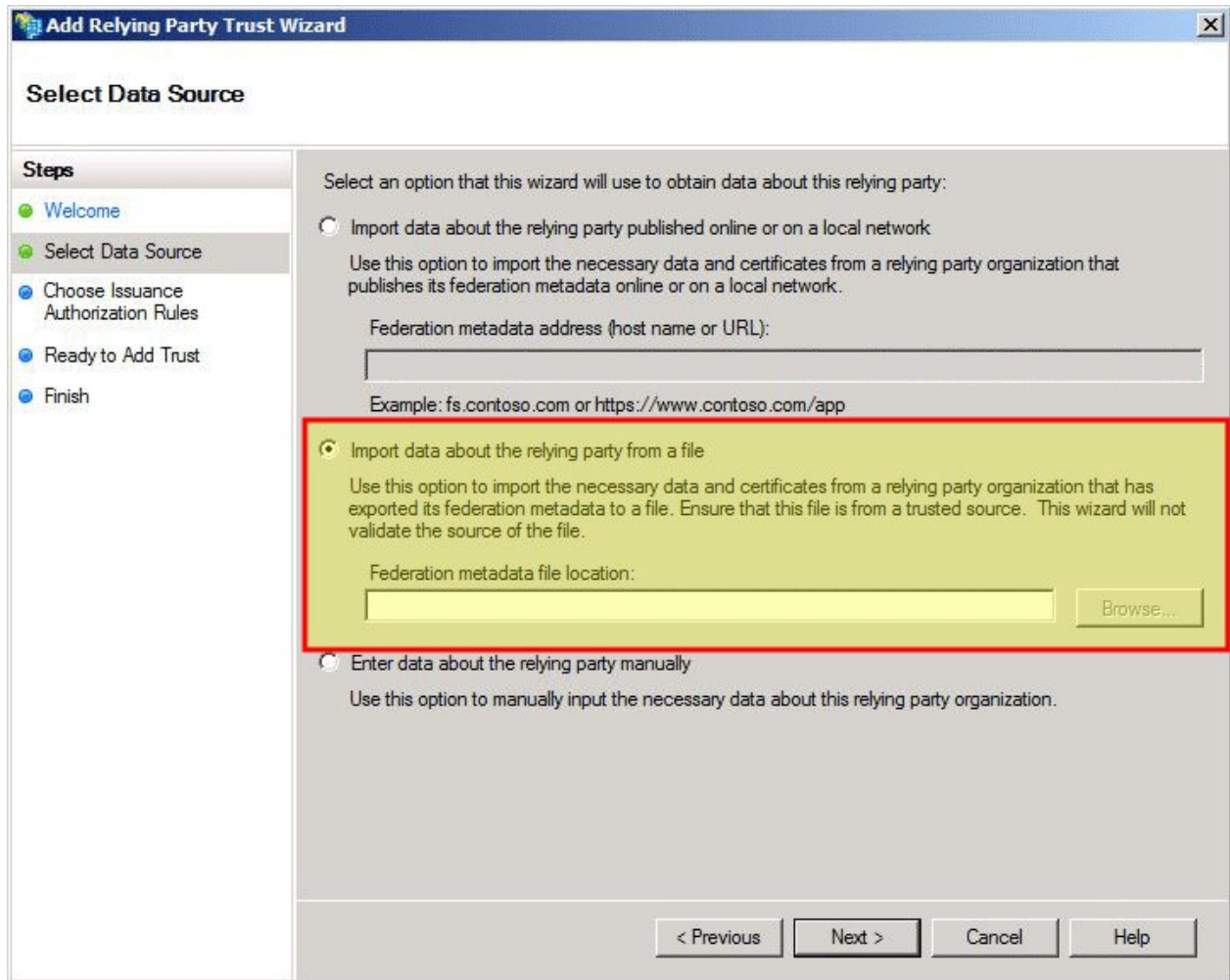
The Add Relying Party Trust Wizard is displayed.



3. Select the **Start** button.

The wizard displays the **Select Data Source** page.

4. If you used Single Sign-On Configuration Utility and generated a SAML metadata file, do the following steps. Otherwise, to configure Microsoft AD FS manually, proceed to step 5.
- Select Import data about the relying party from a file.



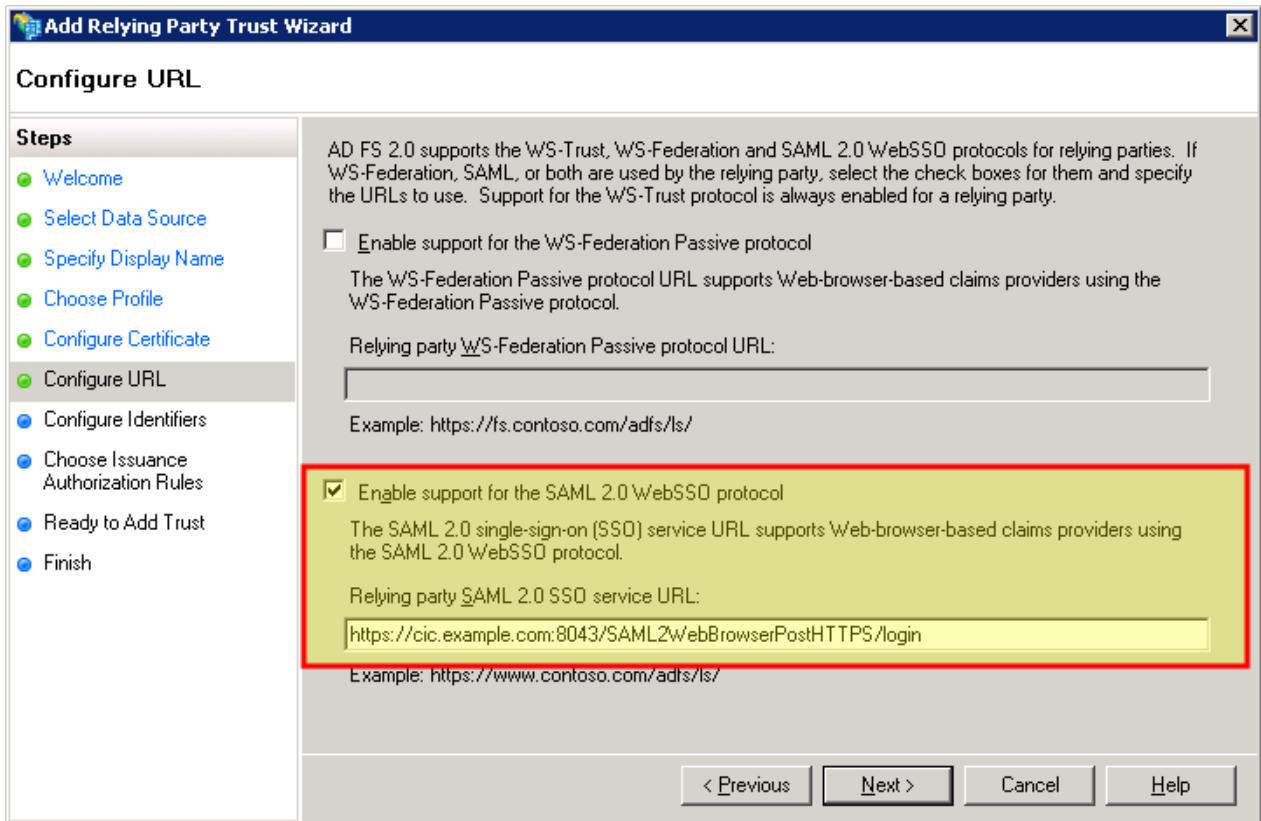
- b. Select the Browse button for the Federation metadata file location box.
- c. In the resulting dialog box, select the SAML metadata file that you created with [Single Sign-On Configuration Utility](#).
- d. Select the Next button.

Microsoft AD FS imports the SAML metadata. You have completed the configuration of Microsoft AD FS for CIC Single Sign-On. Proceed to [Test Single Sign-On for the identity provider](#).

5. Select **Enter data about the relying party manually** and select the **Next** button.

The wizard displays the **Specify Display Name** page.

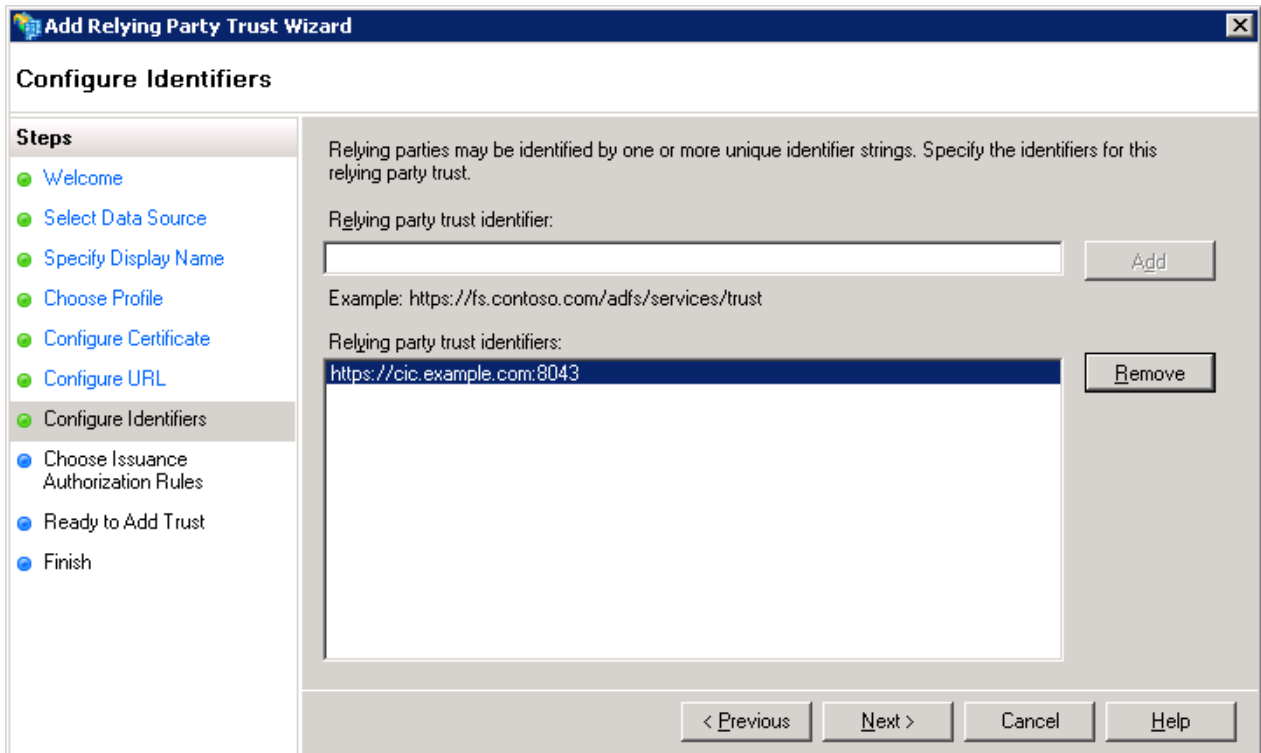
- a. In the **Display Name** box, enter the name of the trust configuration and then select the **Next** button.
The wizard displays the **Choose Profile** page.
- b. Select **AD FS 2.0 profile** and then select the **Next** button.
The wizard displays the **Configure Certificate** page.
- c. Browse to the CIC server trusted certificate and select it.
By default, this certificate is in the following location on the CIC server:
D:\I3\IC\Certificates\Client\Local_Subsystems
This certificate validates signed SAML <AuthnRequest> messages.
- d. On the **Configure Certificate** page, select the **Next** button.
The wizard displays the **Configure URL** page.
- e. Enable the **Enable Support for the SAML 2.0 WebSSO Protocol** check box and specify the [Assertion Consumer Service URL](#).



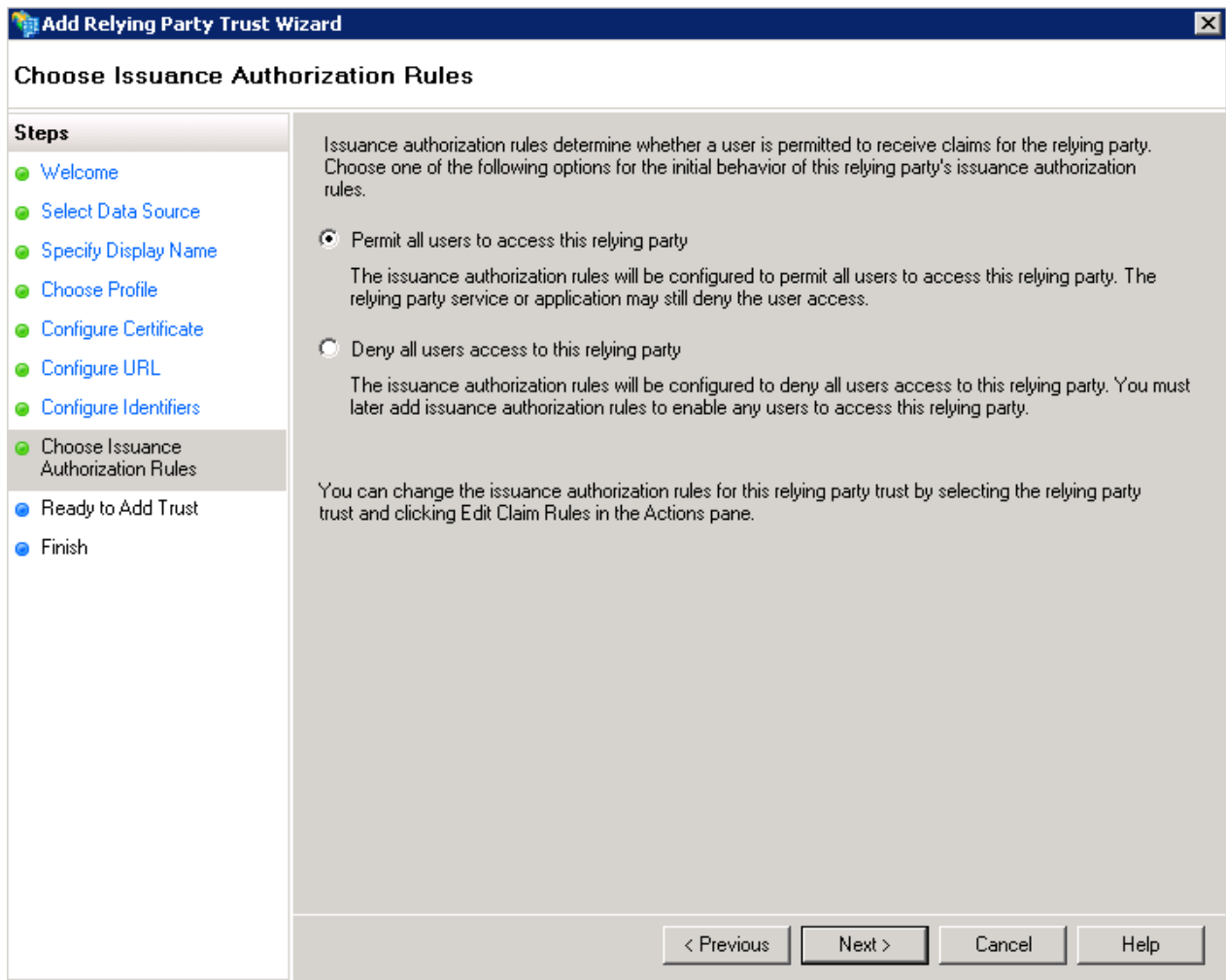
- f. On the **Configure URL** page, select the **Next** button. The wizard displays the **Configure Identifiers** page.
- g. In the **Relying party trust identifier** box, enter the address and port number that you determined in [Determine issuer/provider name/relying party identifier/partner identifier/entity ID](#) and select the **Add** button.

Example:

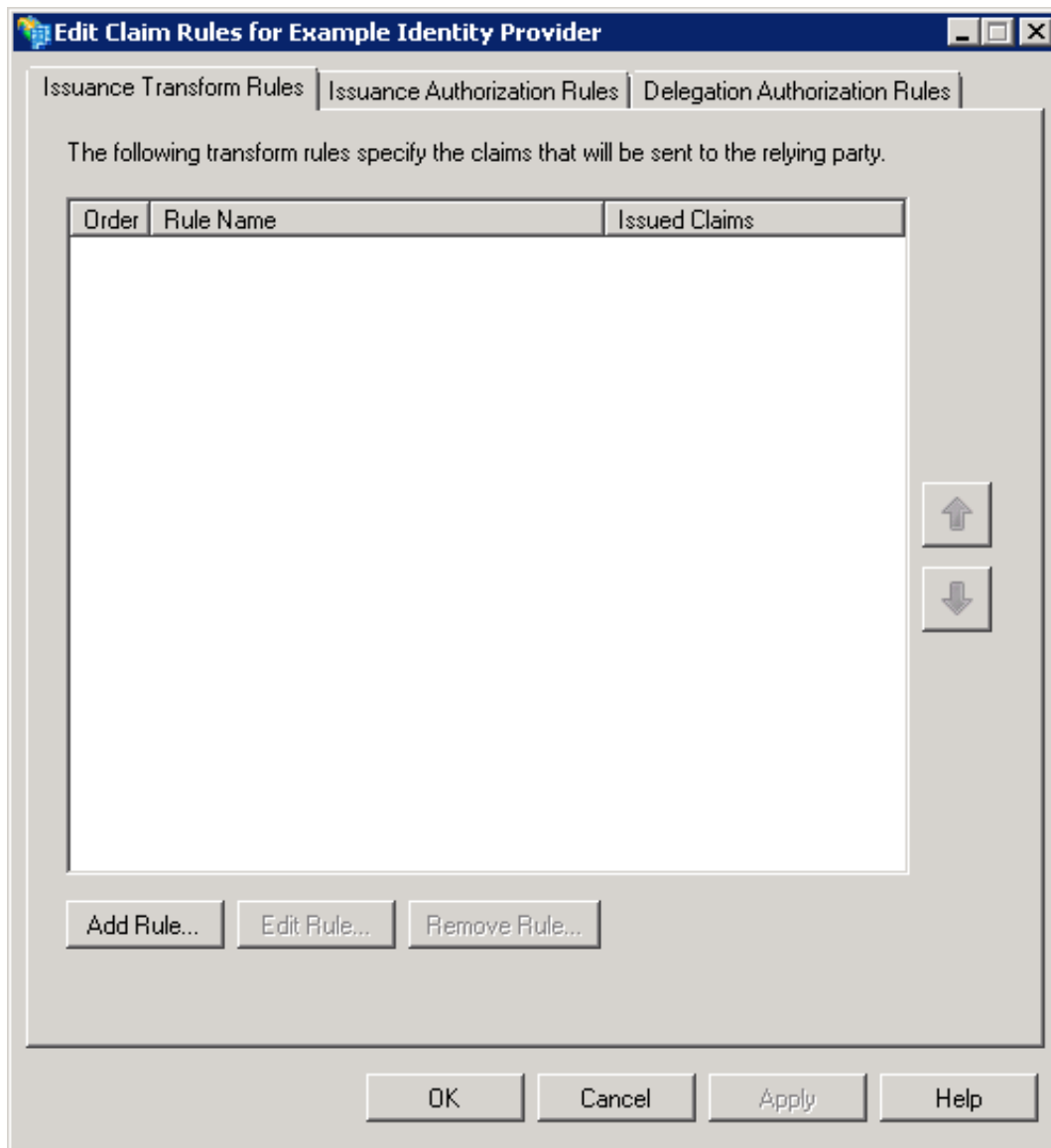
https://cic.example.com:8043



- h. On the **Configure Identifiers** page, select the **Next** button. The wizard displays the **Choose Issuance Authorization Rules** page.

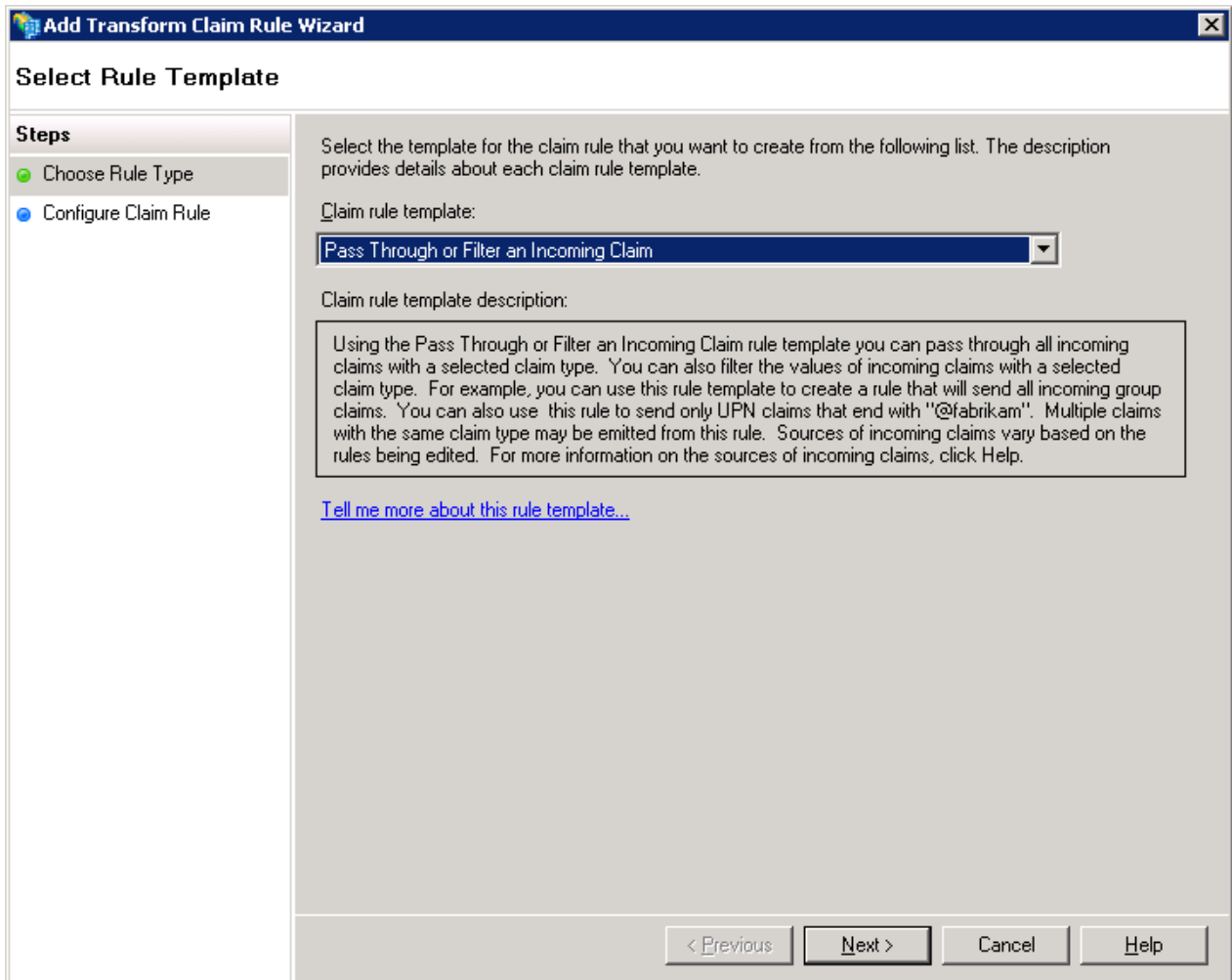


- i. Select one of the appropriate option for either allowing or denying access for all users to the specified relying party.
- j. Select the **Next** button.
The wizard displays the **Ready to Add Trust** page.
- k. Select the **Finish** button.
The **Edit Claim Rules** dialog box is displayed.



I. Select the **Add Rule** button.

The Add Transform Claim Rule Wizard is displayed with the Select Rule Template page.



- m. In the Claim rule template list box, select Pass Through or Filter an Incoming Claim.
 - n. Select the **Next** button.
- The wizard displays the **Configure Rule** page.

Add Transform Claim Rule Wizard

Configure Rule

Steps

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to pass through or filter an incoming claim. You can also configure this rule to filter claims that are generated by previous rules. Specify the claim type and whether only some claim values or all claim values should pass through.

Claim rule name:

Rule template: Pass Through or Filter an Incoming Claim

Incoming claim type:

Incoming name ID format:

Pass through all claim values

Pass through only a specific claim value

Incoming claim value:

Pass through only claim values that end in a specific value:

Ends with:

Example: @fabrikam.com

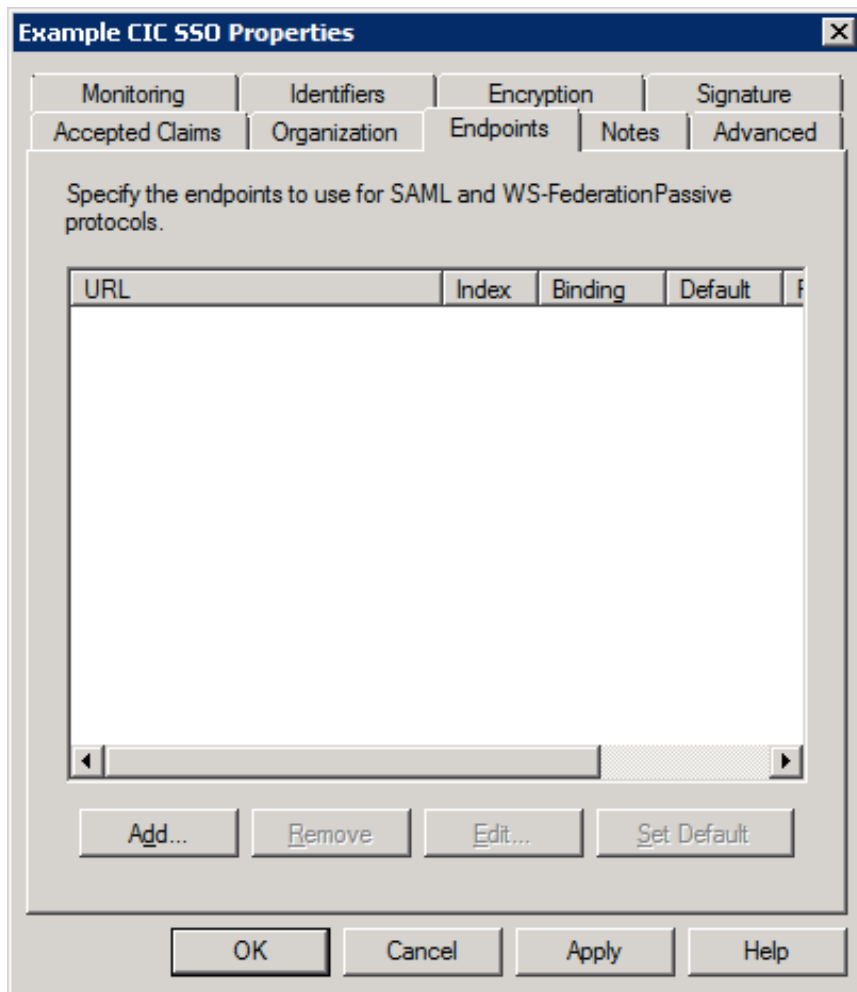
Pass through only claim values that start with a specific value:

Starts with:

Example: FABRIKAM\

< Previous Finish Cancel Help

- o. In the **Claim rule name** box, enter a rule name.
- p. In the **Incoming claim type** list box, select **Windows account name**.
- q. Select the **Finish** button.
- r. In the **Edit Claim Rules** dialog box, select the **OK** button.
- s. In the **AD FS Management** snap-in window, open the properties for the **Relying Party Trust** that you just created.
- t. In the **Properties** dialog box, select the **Endpoints** tab.



- u. Add any additional SAML Assertion Consumer Endpoints for the other Assertion Consumer Service URLs that you will use in your CIC Single Sign-On environment.

Important!

For each ACS URL, increase its index value. The index values must be unique. For more information about ACS URLs, see [Assertion Consumer Service URL](#). For more information on ACS URL index values, see the Microsoft AD FS documentation.

Configure PingOne as an identity provider

Consult the PingOne documentation at <https://documentation.pingidentity.com/pingone/> for configuring the CIC server (service provider) with PingOne (identity provider).

Configure Salesforce as an identity provider

Using Salesforce as your identity provider for a CIC Single Sign-On environment requires you to do the additional configuration procedures in this section.

Important!

Visit the Salesforce help website for updated information on configuring and using Salesforce as a Single Sign-On identity provider.

Test Single Sign-On for the identity provider

After you have either applied the SAML metadata to your identity provider or supplied the SAML metadata to your third-party identity

provider, you can use Single Sign-On Testing Utility to validate that the CIC server, the identity provider, and certificates are configured correctly for Single Sign-On authentication.

1. Start Single Sign-On Testing Utility through one of the following methods:

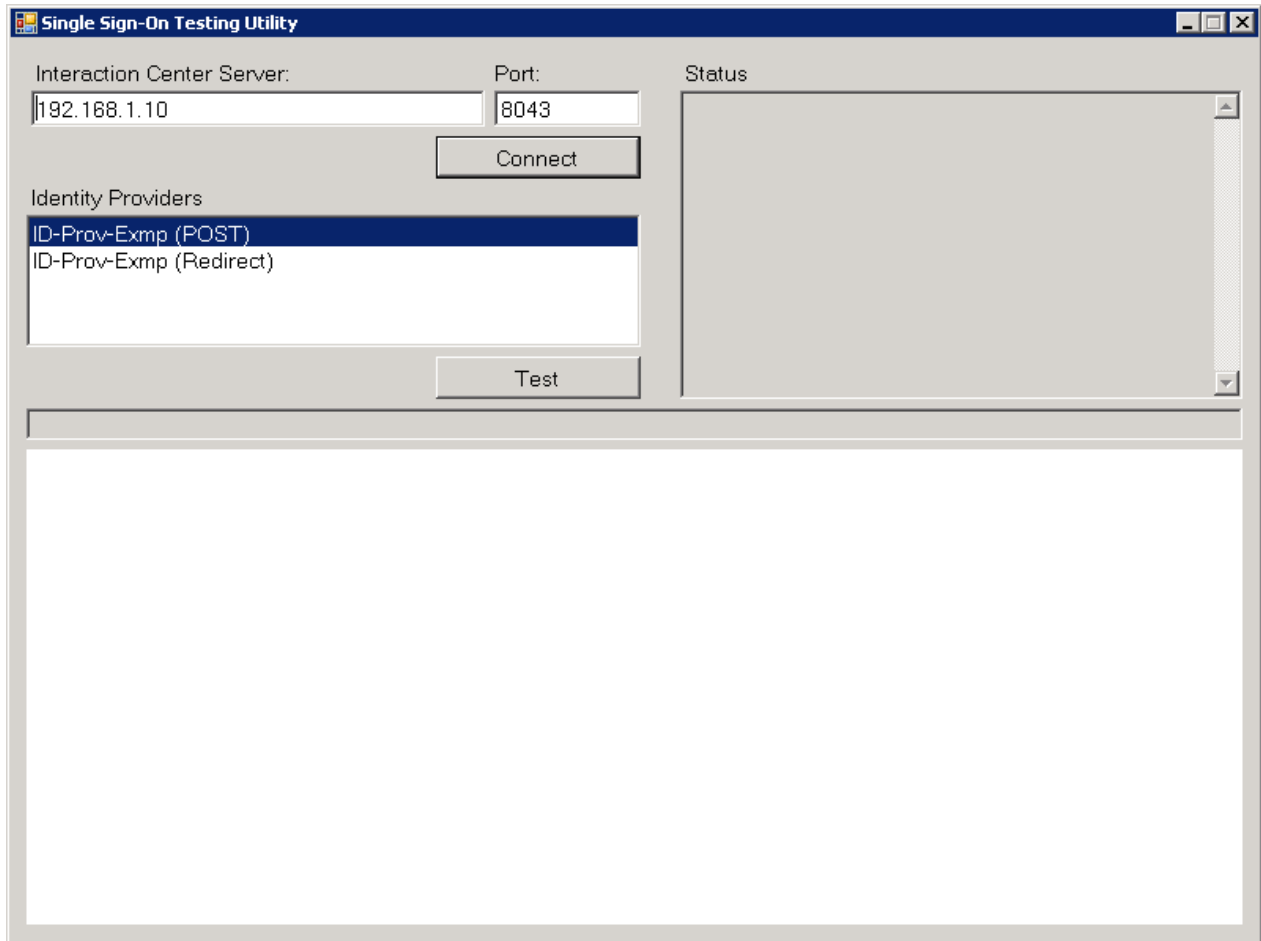
- Run the [Single Sign-On Configuration Utility](#) using the procedure, Use SSO Configuration Utility on a client workstation, and select the **Download Single Sign-On Testing Utility (for Windows)** button.

The Single Sign-On Configuration Utility is downloaded from the CIC server and started.

- Download Single Sign-On Testing Utility from the CIC server. For a default installation of CIC server, the utility is in the following location:

D:\I3\IC\Server\SSOTestingUtility.exe

The Single Sign-On Testing Utility appears.



Note: If you started Single Sign-On Testing Utility through Single Sign-On Configuration Utility, it uses the server address and port information that you provided. If you downloaded Single Sign-On Testing Utility from the CIC server and started it manually, no controls are populated.

2. Enter the CIC server address and the Single Sign-On port number it uses (default: 8043) and select the **Connect** button. If the connection succeeds, the configured identity providers appear in the **Identity Providers** box. If the connection fails, ensure that you entered the correct address for the CIC server and the correct SSO port number that you configured in [Configure Secure Token Server](#).
3. In the **Identity Providers** box, select the identity provider and method to test.
4. Select the **Test** button.

Note: Depending on the settings of the identity provider, you may be prompted to supply your Windows domain credentials. If the test succeeds, a message indicating success appears in the **Status** box:

The Single Sign-On attempt to *<defined identity provider>* was successful!

If the test fails, the **Status** box displays a message that provides indication as to the problem, such as an unresolvable address:

```
System.Net.WebException: The remote name could not be resolved:  
'<defined identity provider>  
at System.Net.HttpWebRequest.GetResponse()  
at SSOTetsingUtility.SSOTestingUtilityForm.TestButton_Click(Object  
p_sender, EventArgs p_event)
```

5. After you have successfully completed the test, ensure that you define any outstanding claims as listed in the success message in the **Status** box.

For more information about claims, see the following content:

- [Identity Provider claims](#) in [Gather identity provider information](#).
- The table in step 19 of [Manually configure identity provider settings](#).

Configure CIC client application workstations

Currently, the following CIC client applications support SAML Single Sign-On:

- Interaction Desktop
- Interaction Connect
- IC Business Manager
- IPA client that is part of IC Server Manager

Note: Notifier-based clients, such as Interaction Attendant, Interaction Administrator, and Interaction Designer, don't support Single Sign-On.

These CIC client applications dynamically present the authentication methods that depend on the ones that you configured in [Enable Single Sign-On authentication on the CIC server](#). You do not need to configure any settings in the CIC client applications to enable Single Sign-On.

Descriptions of the controls associated with Single Sign-On for these CIC client applications are in the help documentation for each application.

Import the HTTPS certificate of the CIC server onto workstations hosting CIC client applications

The only task you must do for supporting a CIC Single Sign-On environment is importing the HTTPS certificate of the CIC server onto the workstations hosting one or more CIC Single Sign-On client applications. It is up to you to determine the best way of importing that certificate onto the workstations. Many administrators use Microsoft Active Directory group policies to import the certificate onto many workstations while other administrators use a manual method for only a few workstations.

Use group policies to import the CIC HTTPS certificate onto workstations

You can also use group policies through Microsoft Active Directory to push the certificate to the workstations with CIC Single Sign-On client applications.

For more information on using group policies in regards to certificates, see Microsoft Windows documentation.

Manual import of the CIC HTTPS certificate onto workstations

In a CIC Single Sign-On environment, CIC client applications must be able to trust the HTTPS certificate that they receive from the CIC server (service provider). To ensure this trust, the Trusted Root Certification Authorities Certificate Store on the workstations that host CIC client applications must have an entry for the Certificate Authority that signed the HTTPS Certificate of the CIC server.

1. Open the following directory location on the CIC server that contains the HTTPS certificate and its encryption keys:
\\13\IC\Certificates\HTTPS
2. Copy the *CICServerName_TrustedCertificate.cer* file to the workstation that hosts a CIC client application.
CICServerName is a variable representing the configured name of your CIC server.
3. Import the *CICServerName_TrustedCertificate.cer* file into the Trusted Root Certificate Authorities Certificate Store of each workstation that hosts a CIC client application that will use the CIC Single Sign-On feature.

Tip: On Windows workstations, you can use Microsoft Management Console (mmc.exe) for administering certificates. For more information about using Microsoft Management Console, see Microsoft Windows documentation.

Test the imported HTTPS certificate

1. After you import the certificate to the workstations, you can test it by doing the following steps on the workstation:
 - a. Open a web browser.
 - b. Navigate to **https://CICServerAddress:8043**
CICServerAddress is a variable representing the address of your CIC server.
If the web browser displays no warning or prompt, the imported HTTPS certificate for the CIC server is functioning.
2. If the target URL of the identity provider uses HTTPS (Secure HTTPS), repeat this procedure to copy and import the identity provider validation certificate that you received from the identity provider in [Gather identity provider information](#).

Troubleshooting

This section contains procedures and information for troubleshooting and correcting problems with your Customer Interaction Center Single Sign-On environment.

Single Sign-On troubleshooting tools

For troubleshooting Single Sign-On issues, Genesys recommends Fiddler by Telerik. Fiddler is a web debugging proxy that supports all major operating systems and many frameworks, such as Java, Ruby, and Microsoft .NET.

You can download Fiddler from the following website: <http://www.telerik.com/fiddler>

Install and configure Fiddler by following the instructions on the following website: <http://docs.telerik.com/fiddler/configure-fiddler/tasks/configurefiddler>

Caution!

You should allow the interception and decryption of HTTPS traffic on test workstations only. Genesys is not responsible if you allow these activities on a production workstation and information is exposed.

Use the Fiddler online documentation to ensure that you configure Fiddler as required for your specific CIC Single Sign-On environment, including SAML profiles and bindings.

You can then use Fiddler to capture and view web traffic for troubleshooting where the Single Sign-On configuration is failing.

Examine log files

If a CIC client application based on the .NET Framework has trouble authenticating in your CIC Single Sign-On environment, check the application logs on the Windows host computer for **IceLib** traces. Specifically, you want to examine entries containing class names that start with **SAML2**.

For the CIC server, you can view the **ic_sts** logs.

If you are using Microsoft Active Directory Federated Services (AD FS), you can examine the Windows Event Log for identity provider items.

Any IC Web Services errors are displayed to users in the web browser during authentication.

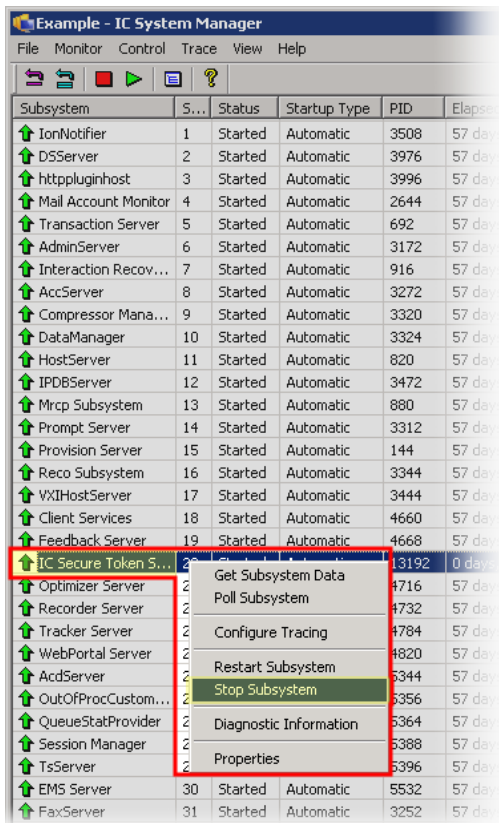
Updating the CIC server causes certificate validation issues

If you upgraded your CIC server from one of the following versions, the upgrade to the current version creates an incompatibility with your previous HTTPS certificates:

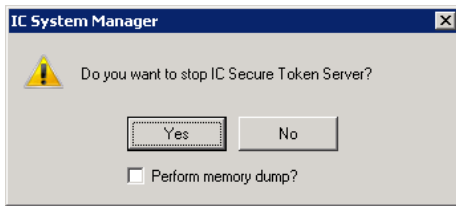
- CIC 4.0 SU5
- CIC 4.0 SU6
- CIC 2015 R1

To correct this issue, do the following steps:

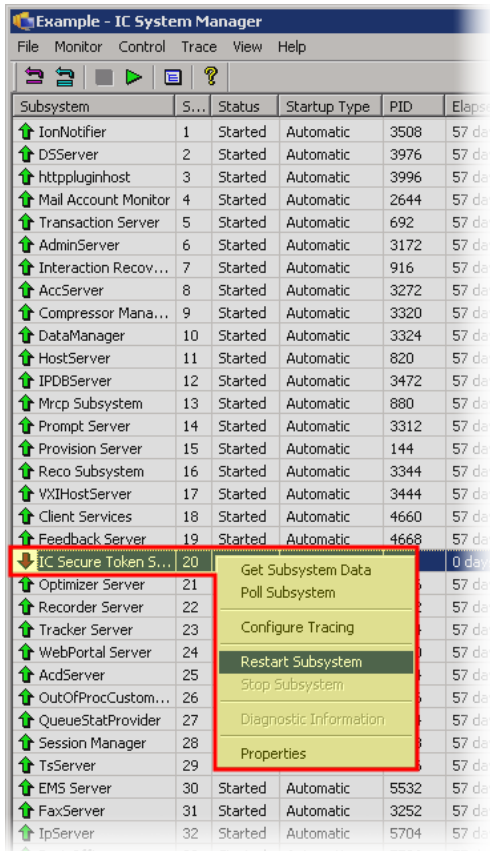
1. Open **IC System Manager** for the affected CIC server.
2. In the **Subsystem** column of the **IC System Manager** window, locate the **IC Secure Token Server** entry.
3. Right-click the **IC Secure Token Server** entry and select **Stop Subsystem** from the shortcut menu.



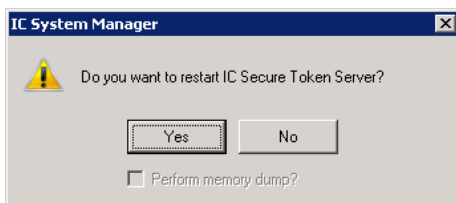
4. When IC Server Manager prompts you to confirm that you want to stop the **IC Secure Token Server** service, click **Yes**.



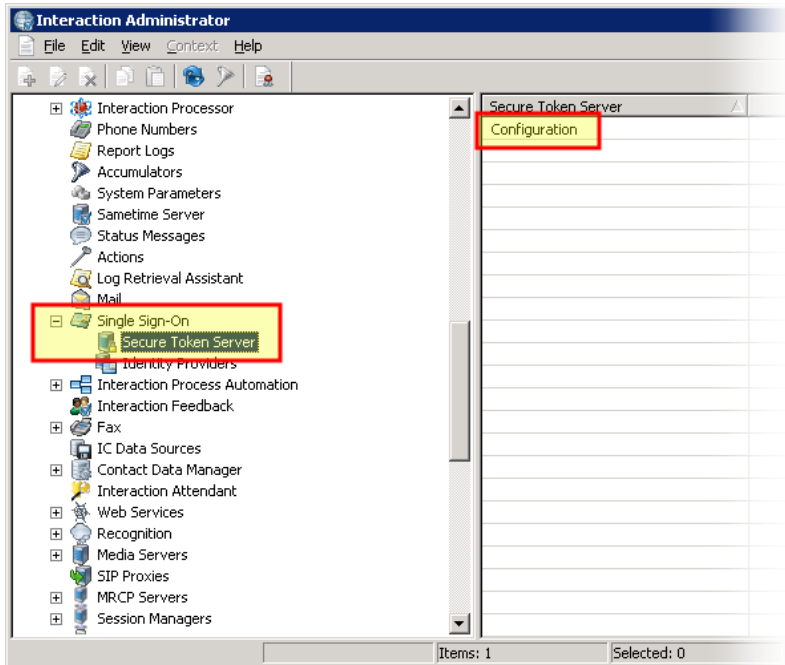
5. Wait for the **IC Secure Token Server** subsystem to stop, which is then indicated with a downward-pointing, red arrow.
6. On the CIC server, open a **Command Prompt** window as an Administrator.
7. In the **Command Prompt** window, navigate to the drive where the CIC server software was installed by entering and executing the following command:
D:
D: is the default drive on which the CIC server software is installed. If you installed the CIC server software to a different drive, replace D with the appropriate letter.
8. Navigate to the Default subdirectory by entering and executing the following command:
cd \I3\IC\Certificates\ICSecureTokenServer\Default
9. Rename the existing Secure Token Server HTTPS certificate by entering and executing the following command:
ren ICSecureTokenServer*. ICSecureTokenServer*.backup
10. Navigate to the HTTPS subdirectory by entering and executing the following command:
cd ..\HTTPS
11. Copy the previous private key of the CIC server to the ICSecureTokenServer subdirectory by entering and executing the following command:
copy *CICServerName*_PrivateKey.bin ..\ICSecureTokenServer\Default\ICSecureTokenServerPrivateKey.bin
CICServerName is a variable representing the name of this CIC server.
12. Copy the previous HTTPS certificate of the CIC server to the ICSecureTokenServer subdirectory by entering and executing the following command:
copy *CICServerName*_Certificate.cer ..\ICSecureTokenServer\Default\ICServerTokenServerCertificate.cer
CICServerName is a variable representing the name of this CIC server.
13. Close the **Command Prompt** window by entering and executing the **exit** command.
14. In the **IC System Manager** window, restart the **IC Secure Token Server** entry by right-clicking the entry and selecting **Start** from the resulting shortcut menu.



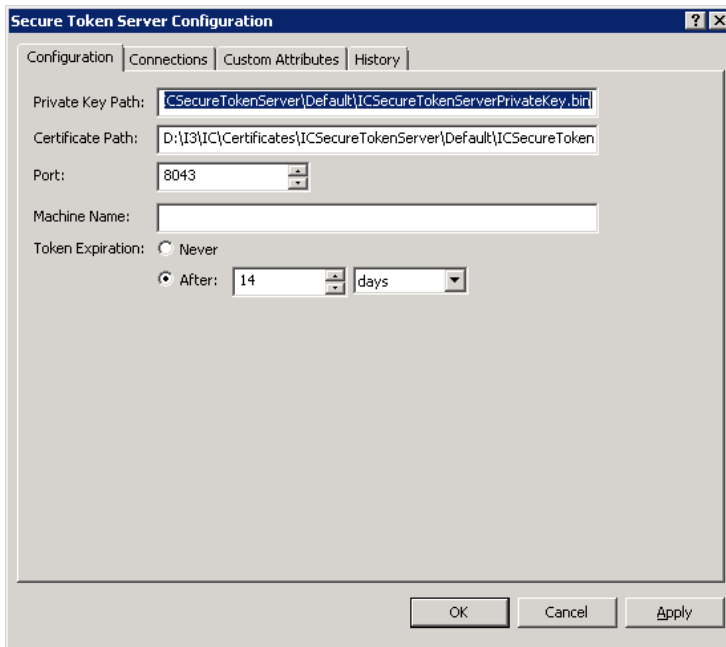
15. When IC Server Manager prompts you to confirm that you want to restart the **IC Secure Token Server** service, select the **Yes** button.



16. Wait for the **IC System Manager** window to show a green, upward-pointing arrow for the **IC Secure Token Server** service to indicate that it has successfully restarted.
17. Close IC System Manager.
18. Open Interaction Administrator for the CIC server.
19. In the left pane of the **Interaction Administrator** window, expand the **System Configuration > System Sign-On** container and then select the **Secure Token Server** item.
20. In the right pane, double-click the **Configuration** item.



Interaction Administrator displays the Secure Token Server Configuration dialog box.



21. Ensure that the paths specified in the **Private Key Path** and **Certificate Path** boxes reference the files you created in steps 11 and 12, respectively.
22. Test a Single Sign-On logon attempt on the CIC client application to ensure that functionality has been restored.

SAML 2.0 message exchange example

This section provides an example of a standard SAML 2.0 <AuthnRequest> message and the <AuthnResponse> from the identity provider. The Secure Token Server subsystem of the CIC server expects identity providers to provide their authentication results in this type of response.

This example uses the HTTP POST binding. In this example, the identity provider uses the following URL address:

<https://cic.example.com/adfs/ls/>

1. The user agent accesses the Secure Token Server endpoint for authentication:

```
GET /SAML2WebBrowserPOSTHTTPS/login?
HTTP/1.1
Host: 10.8.3.198:8043
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101
Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

2. The secure token server returns the SAML <AuthnRequest> message that the user agent sends to the identity provider:

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/html
```

```

Date: Fri, 03 Jan 2014 19:43:12 GMT
Server: I3STSCore
Content-Length: 1616
<html>
<head>
<title>SAML2WebBrowserPOSTHTTPS</title>
</head>
<body>
Authenticating with the identity provider ...
<form method="POST" action="https://cic.example.com/adfs/ls/">
<input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkFldGhuUmVxdWVzdCAGICB4bWxuczpzYWlsPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIiAgICB4bWxuczpzYWlsD01dXJ0aW9uOnc2Lz0m5hbWVzOnRjO1NBTUw6M14wOnByb3RvY29s:

<input type="hidden" name="RelayState"
value="12F6b8be-72e9-4bb9-97a1-4392f2746ecd"/>
<noscript>
<p>JavaScript is disabled. Click Submit to continue.</p>
<input type="submit" value="Submit"/>
</noscript>
</form>
<script language="javascript">
window.setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>

```

The Base64 encoded SAML <AuthnRequest> message for step 2, when decoded, is as follows:

```

<saml:AuthnRequest xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="4df85849a56233459ba4acf672ffa53d"
Version="2.0" IssueInstant="2014-01-03T19:43:12.337249Z"
Destination="https://cic.example.com/adfs/ls/ Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
ForceAuthn="false" IsPassive="false" AssertionConsumerServiceURL="https://10.8.3.198:8043/SAML2WebBrowserPOSTHTTPS/login"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
ProviderName="https://10.8.3.198:8043">
<saml:Issuer>https://10.8.3.198:8043</saml:Issuer>
</saml:AuthnRequest>

```

3. The user agent posts the SAML <AuthnRequest> message to the identity provider:

```

POST /adfs/ls/ HTTP/1.1
Host: cic.example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101
Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://10.8.3.198:8043/SAML2WebBrowserPOSTHTTPS/login?
Cookie: MSISIPSelectionPersistent=aHR0cDovL0NsYXkubWk2LnNvbS9hZGZzL3N1cnZpY2VzL3RydXN0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 936

SAMLRequest=PHNhbWxwOkFldGhuUmVxdWVzdCAGICB4bWxuczpzYWlsPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIiAgICB4bWxuczpzYWlsD01dXJ0aW9uOnc2Lz0m5hbWVzOnRjO1NBTUw6M14wOnByb3RvY29s:
72e9-4bb9-97a1-4392f2746ecd

```

4. The identity provider validates the user and responds with a SAML <AuthnResponse> message:

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/7.5
X-AspNet-Version: 2.0.50727
Set-Cookie: MSISSamlRequest=; expires=Thu, 02-Jan-2014 19:43:14
GMT; path=/adfs/ls
Set-Cookie: MSISAuthenticated=MS8zLzIwMTQgNzo0MzoxNCBQTQ==;
path=/adfs/ls; secure; HttpOnly
Set-Cookie: MSISLoopDetectionCookie=MjA0NzUwMS0wMzoxOT00MzoxNEpccMQ==;
path=/adfs/ls; secure; HttpOnly
Persistent-Auth: true
X-Powered-By: ASP.NET
Date: Fri, 03 Jan 2014 19:43:14 GMT
Content-Length: 5580

<html>
<head>
<title>Working...</title>
</head>
<body>
<form method="POST" name="hiddenform"
action="https://10.8.3.198:8043/SAML2WebBrowserPOSTHTTPS/login">
<input type="hidden" name="SAMLResponse"
value="PHNhbWxwOkFldGhuUmVxdWVzdCAGICB4bWxuczpzYWlsPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIiAgICB4bWxuczpzYWlsD01dXJ0aW9uOnc2Lz0m5hbWVzOnRjO1NBTUw6M14wOnByb3RvY29s:

/>
<input type="hidden" name="RelayState"
value="12F6b8be-72e9-4bb9-97a1-4392f2746ecd" />
<noscript>
<p>
Script is disabled. Click Submit to continue.
</p>
<input type="submit" value="Submit"
/>
</noscript>
</form>
<script language="javascript">
window.setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>

```

The Base64 encoded SAML <AuthnResponse> message for step 4, when decoded, is as follows:

```
<samlp:Response ID=" 4d9e1e67-7c5c-464e-91ad-78738410db53"
Version="2.0" IssueInstant="2014-01-03T19:43:14.391Z"
Destination="https://10.8.3.198:8043/SAML2WebBrowserPOSTHTTPS/login"
Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
InResponseTo=" 4df85849a56233459ba4acf672ffa53d" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
<Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
  http://cic.example.com/adfs/services/trust
</Issuer>
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"
/>
</samlp:Status>
<Assertion ID=" a4e99a02-4123-45ff-b9d5-7cad3d650220"
IssueInstant="2014-01-03T19:43:14.391Z" Version="2.0"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
<Issuer>
  http://cic.example.com/adfs/services/trust
</Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
/>
<ds:Reference URI="# a4e99a02-4123-45ff-b9d5-7cad3d650220">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256"
/>
<ds:DigestValue>
  +34wpDamiPwpnOoP231cvBDX6tKT/4hLzBKw7Zc56ks=
</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
  d1PPkEb6RzAkzbn9nQBVNqqt15MbR1RJ/eJThr/5YiryWCvPmX3MfQ8lqK1VC6ovEeLdmcnRGJ7C/29SGGIUUb10X830W7Bd4N4qfS25XICVpj23NT3zwEY1Cb2CCZJThueGFpG31IoTffLqfoDKDi7utyInZ4TNOJpLkLJ79yEJ
</ds:SignatureValue>
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>
  MIIIC1DCCAbvgAwIRAgIQFwiR+G4E0r1HK/afUzsj/DANBqkqhkiG9w0BAQsFADAmMSQwIgwYDVQQDEExtBREZTFIENpZ25pbmcgLSBDbGF5Lm1pL15jb20wHhcNMjAwODA4MjYzODI1WhcNMjAwODA4MjYzODI1WjAmMSQwIgwYDVQQI
</ds:X509Certificate>
</ds:X509Data>
</KeyInfo>
</ds:Signature>
<Subject>
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<SubjectConfirmationData InResponseTo=" 4df85849a56233459ba4acf672ffa53d"
NotOnOrAfter="2014-01-03T19:48:14.391Z" Recipient="https://10.8.3.198:8043/SAML2WebBrowserPOSTHTTPS/login"
/>
</SubjectConfirmation>
</Subject>
<Conditions NotBefore="2014-01-03T19:43:14.390Z"
NotOnOrAfter="2014-01-03T20:43:14.390Z">
<AudienceRestriction>
<Audience>
  https://10.8.3.198:8043
</Audience>
</AudienceRestriction>
</Conditions>
<AttributeStatement>
<Attribute Name="http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname">
<AttributeValue>
  MI6\Administrator
</AttributeValue>
</Attribute>
</AttributeStatement>
<AuthnStatement AuthnInstant="2014-01-03T19:43:14.358Z">
<AuthnContext>
<AuthnContextClassRef>
  urn:federation:authentication:windows
</AuthnContextClassRef>
</AuthnContext>
</AuthnStatement>
</Assertion>
</samlp:Response>
```

5. The user agent posts the SAML <AuthnResponse> message to the secure token server:

```
POST /SAML2WebBrowserPOSTHTTPS/login
HTTP/1.1
Host: 10.8.3.198:8043
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101
Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://cic.example.com/adfs/ls/auth/integrated/
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 5155

SAMLResponse=PHNhbWw0Lj1c3BvbW50Lj1EIElEPSJfNGQ5ZTF1NjctN2M1Yy00NjRlLTxxYWQtNzgzMzgzMTBkYjUzIiEiBwZkZzaW9uPSIyLjAiIEI1zc3V1SW5zdGFudD0iMjAwODA4MjYzODI1WhcNMjAwODA4MjYzODI1WjAmMSQwIgwYDVQQDEExtBREZTFIENpZ25pbmcgLSBDbGF5Lm1pL15jb20wHhcNMjAwODA4MjYzODI1WhcNMjAwODA4MjYzODI1WjAmMSQwIgwYDVQQI72e9-4bb9-97a1-4392f2746ecd
```

6. The secure token server issues the user agent an ININ-STS-TOKEN:

```
HTTP/1.1 200 OK
ININ-STS-TOKEN: (3"ExpirationTime"20140117194314.989423,"IdentityProvider"12f6b8be-72e9-4bb9-97a1-4392f2746ecd", "AuthenticationType""SAML2WebBrowserPOSTHTTPS") | Js4diZHLKcW88R60KkC149udu1zSkNRKmxKb6P0FphF9jZ/gudeZ4gQ+4kOztj40vYELLeLJYuwSwGS+F4TzGbxJOMc3Wkq8s3c02QwaBVxgWPT:
```

Date: Fri, 03 Jan 2014 19:43:15 GMT
Server: I3STSCore
Content-Length: 0

Change Log

The following changes have been made to this document since release.

Date	Changes
08-February-2014	Initial publication
10-September-2014	<ul style="list-style-type: none"> Updated documentation to reflect changes required in the transition from version 4.0 SU 6 to CIC 2015 R1, such as updates to product version numbers, system requirements, installation procedures, references to Product Information site URLs, and copyright and trademark information. Added a section called Configuration notes for ADFS 2.0.
23-February-2015	Added information about protocols and configuration. Updated for latest releases of software.
26-March-2015	Added information to Appendix A about configuring Salesforce.
13-April-2015	Added information about wizard to configure identity providers. Updated some figures and made minor tweaks.
06-May-2015	In Chapter 2, added note about Shibboleth identity provider and added a new section, "Determining the Assertion Consumer Service URLs." Divided Appendix A into three separate appendices (A,B,C) and expanded the coverage of ADFS configuration in the new Appendix A.
28-September-2015	<ul style="list-style-type: none"> Documented reformatted for corporate rebranding Updated legal page IC-128901 - Added information regarding support of new feature for importing SAML 2.0 metadata from identity providers Added conceptual content in a new introductory section Created additional procedures Provided more details and examples in procedures
05-November-2015	<ul style="list-style-type: none"> Updated procedures for generating or importing HTTPS certificates on the CIC server for use in Single Sign-on for various configurations IC-131188 DocLink: Custom Forms support for SSO - Adjust content to reflect support of web-based HTML form authentication for IceLib-based CIC clients Implemented general edits and clarifications
29-November-2016	<ul style="list-style-type: none"> Updated incorrect screenshot to reflect web-form authentication support in Single Sign-On configuration dialog box in Interaction Administrator. Updated Copyright and Trademark Information page
16-March-2018	Rebranded to Genesys.
21-May-2019	Reorganized the content only, which included combining some topics and deleting others. For more details, see CICDOC-189 .
27-February-2020	<ul style="list-style-type: none"> Clarified that the IPA client that is part of IC Server Manager (and not IC Server Manager itself) supports Single Sign-On. Added a note that Notifier-based clients don't support Single Sign-On. <p>For more information, see Configure CIC client application workstations.</p>