HP Select Identity Software

Connector for Windows® Active Directory (Bidirectional LDAP Based)

Connector Version: 2.10

Installation and Configuration Guide

Document Release Date: September 2007
Software Release Date: September 2007

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils
- Commons-collections
- Commons-logging
- Commons-digester
- Commons-httpclient
- Element Construction Set (ecs)
- Jakarta-poi
- Jakarta-regexp
- Logging Services (log4j)

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge
- iText (for JasperReports) developed by SourceForge
- BeanShell
- Xalan from the Apache XML Project
- Xerces from the Apache XML Project
- Java API for XML Processing from the Apache XML Project
- SOAP developed by the Apache Software Foundation
- JavaMail from SUN Reference Implementation
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation
- Java Cryptography Extension (JCE) from SUN Reference Implementation
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation

- OpenSPML Toolkit from OpenSPML.org
- JGraph developed by JGraph
- Hibernate from Hibernate.org
- BouncyCastle engine for keystore management, bouncycastle.org

Trademark Notices

Support

You can visit the HP software support web site at:

http://www.hp.com/go/hpsoftwaresupport

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

1 Documentation Map

This chapter describes the organization of HP Select Identity connector documentation and provides necessary information about how to use the documentation set to install and configure the connectors.

Figure 1 illustrates the documentation map for HP Select Identity connector. For a list of available product documentation, refer to the Table 1.

Figure 1 Documentation Map

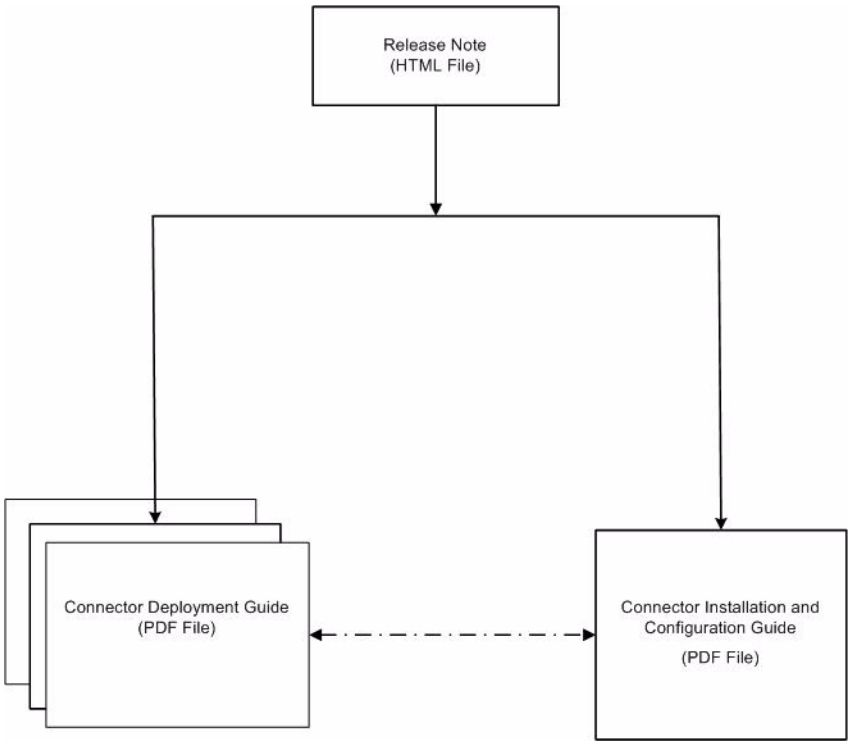


Table 1 Connector Documentation

Document Title and Filename	Contents	Location
Release Note `Active Directory BiLDAP Connector v2.10 Release Note.htm`	This file contains necessary information on new features of the connector, enhancements, known problems or limitations, and support information.	`/Docs/` subdirectory under the connector directory.
Connector Deployment Guide (for Select Identity 4.20 `connector_deploy_SI4.20.pdf`	Connector deployment guides provide detailed information on: • Deploying a connector on an application server. • Configuring a connector with Select Identity. Refer to these guides when you need generic information on connector installation.	`/Docs/` root directory on the product's CD media.
Connector Deployment Guide (for Select Identity 4.10-4.13) `connector_deploy_SI4.13.pdf`		
Connector Deployment Guide (for Select Identity 4.0/4.01) `connector_deploy_SI4.pdf`		
Connector Installation and Configuration Guide `Active Directory BiLDAP_guide.pdf`	Connector installation and configuration guide provides installation instructions for a specific connector. It contains resource specific configuration details.	`/Docs/` subdirectory under the connector directory.

2 Introduction

This chapter gives an overview of the HP Select Identity connector for Active Directory. An HP Select Identity connector for Active Directory enables you to provision users and manage identities on Active Directory. At the end of this chapter, you will be able to know about:

- The benefits of HP Select Identity.
- The role of a connector.
- The connector for Active Directory.

About HP Select Identity

HP Select Identity provides a new approach to identity management. Select Identity helps you automate the process of provisioning and managing user accounts and access privileges across platforms, applications, and corporate boundaries. Select Identity communicates with the enterprise information system through connectors, and automates the tasks of identity management. The enterprise information system, which is also referred to as **resource**, can be a database, a directory service, or an ERP package, among many others.

About Connectors

You can establish a connection between a resource and Select Identity by using a connector. A connector is resource specific. The combination of Select Identity and connector helps you perform a set of tasks on the resource to manage identity. A connector can be **unidirectional** or **bidirectional**. A unidirectional connector helps you manage identities from Select Identity, but if any change takes place in resource, it cannot communicate that back to Select Identity. On the other hand, a bidirectional connector can reflect the changes made on resource back to Select Identity. This property of bidirectional connectors is known as **reverse synchronization**.

About Active Directory Bidirectional LDAP Connector

The bidirectional LDAP connector for Microsoft Active Directory — hereafter referred to as Active Directory Bidirectional LDAP connector — enables Select Identity to perform the following tasks in Active Directory server:

For `user` objectClass:

- Add, update, and remove users
- Retrieve user attributes

- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords
- Expire user passwords
- Retrieve all entitlements
- Retrieve a list of supported user attributes
- Grant and revoke entitlements to and from users
- User rename (Change of CN attribute)
- User move across OUs in the same domain
- Multi-domain features:
  - Support for AD forest: User forward provision to any domain in a multi-domain AD forest
  - Support for multiple domain controllers (DCs) and global catalogs (GCs) in the AD forest
  - Assign and un-assign user to/from any group (entitlement) in multi-domain forest
  - User change detection (add, delete, rename, profile modify, link/unlink, reset password, move cross OU or domain) from all the domains in the AD forest
- Failover features:
  - Forward provision failover support. Try secondary DC/GC if the primary DC/GC failed (depends on the operation type).
  - Retry if fails on domain controller in both forward and reverse polling. Number of retries is configurable.

For `contact` objectClass:

- Add, update, and remove contacts
- Retrieve contact attributes
- Grant and revoke entitlements to and from contacts

For definitions about core concepts of Active Directory Domain Services, such as forest, domain, and global catalog, visit Microsoft MSDN website at:

http://msdn2.microsoft.com/en-us/library/aa772157.aspx

Other features:

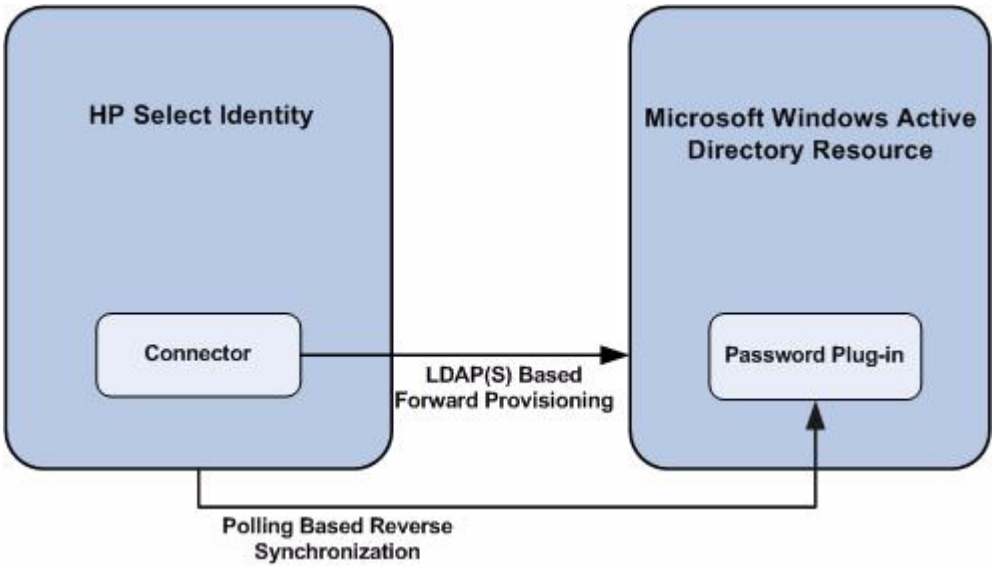
- Supports Select Identity Connector Interface 4.x
- Supports mutual authentication
- Supports moving user across domain on Windows 2000 native mode and Windows 2003 Server
- Supports Select Identity username change
- Supports multi-valued attributes for mutil-valued AD attributes
- Supports both 32bit and 64bit AD server
- Supports both Parent-Child and Peer-to-Peer forest environments

High-Level Architecture

Figure 2 illustrates a high-level architecture of Active Directory Bidirectional LDAP connector. This is a bidirectional, Lightweight Directory Access Protocol Version 3 (LDAPv3) compliant connector that pushes changes made to user data in the Select Identity database to a target Active Directory server. The connector uses the Java LDAP Application Program Interfaces (APIs) to provision users and their entitlements in the LDAP server, which in turn pushes the data to the Active Directory server.

A reverse synchronization feature reconciles user account changes made on the Active Directory resource with Select Identity. Select Identity periodically polls the Active Directory resource to retrieve changes through the connector.

Figure 2 High-Level Architecture of the Active Directory Bidirectional LDAP Connector



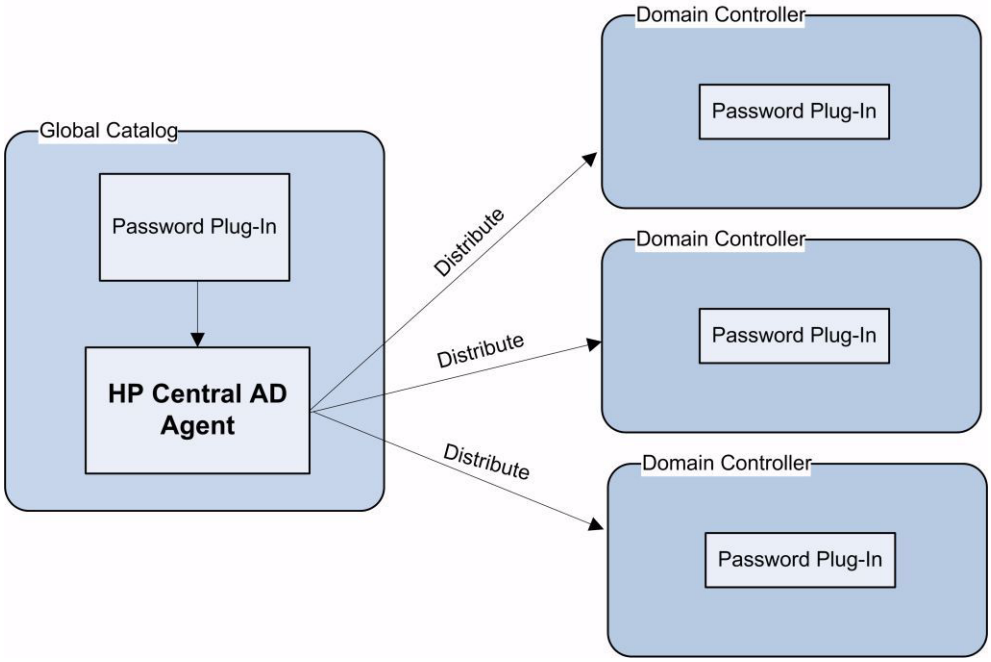
This connector can be used with all versions of Select Identity (4.0-4.20).

Password Plug-In

The Password Plug-In captures the password changes in Active Directory and stores the changed password in encrypted form on Active Directory system. The change is picked up by the connector during next polling operation. This agent only updates Active Directory and does not directly interact with Select Identity web service. The Password Plug-In is optional and if it is not installed, password changes will not be reconciled to Select Identity.

In an Active Directory multi-domain forest environment, the Password Plug-In can be distributed onto all Domain Controller servers by running HP Central AD Agent setup utility.

Figure 3 Architecture of HP Central AD Agent



Overview of Installation Tasks

Before you start installing the connector, you must ensure that system requirements and all the installation prerequisites are met. Refer to the Table 2 for an overview of installation tasks.

Table 2 Organization of Tasks

Task Number	Task Name	Reference
1	Install the connector on the Select Identity server.	See Installing the Connector on page 15.
	— Meet the system requirements.	See System Requirements on page 16.
	— Perform the pre-installation tasks: Install Active Directory certificate on the application server hosting Select Identity.	See Pre-Installation Tasks on page 16.
	— Extract contents of the Schema file (file that contains the mapping files for the connector) to a location on the Select Identity server.	See Extracting Contents of the Schema File on page 32.
	— Verify configurable parameters in the `ActiveDirconfig.poperties` file.	See Verifying Configurable Parameters on page 32.
	— Install the Resource Adapter Archive (RAR) of the connector on an application server.	See Installing the Connector RAR on page 38.
	— Configure Select Identity database to block cyclic request.	See Configuring the Database on Select Identity System to Block Cyclic Request on page 38.
2	Install agent module for Active Directory Bidirectional LDAP connector.	See Installing Agent on page 41.
	— Install Password Plug-In	See Installing Password Plug-In on page 41.
	— Distribute Password Plug-In	See Distributing Password Plug-In on page 48.

Table 2 Organization of Tasks (cont'd)

Task Number	Task Name	Reference
3	Configure the connector with the Select Identity server.	See Configuring the Connector with Select Identity on page 51.
	— Add a new connector to Select Identity.	See Add a New Connector on page 51.
	— Add a new resource to Select Identity.	See Add a New Resource on page 51.
	— Map Active Directory attributes to Select Identity attributes.	See Map Attributes on page 56.
	— Configure Workflow External Call.	See Configure Workflow External Call on Select Identity on page 59.

3 Installing the Connector

This chapter elaborates the procedure to install Active Directory Bidirectional LDAP connector on Select Identity server. At the end of this chapter, you will know about

- Software requirements to install the Active Directory Bidirectional LDAP connector.
- Prerequisite conditions to install Active Directory Bidirectional LDAP connector.
- Procedure to install Active Directory Bidirectional LDAP connector.

Active Directory Bidirectional LDAP Connector Files

The Active Directory Bidirectional LDAP connector is packaged in the following files, which are located in the `Bidirectional LDAP Connector - Active Directory` folder on the Select Identity Connector CD:

Table 3 Active Directory Bidirectional LDAP Connector Files

Serial Number	File Name	Description
1	For Select Identity 4.0-4.13: • `ActiveDirConnector.rar`	They contain the binaries for the connector.
	For Select Identity 4.20: • `ActiveDirConnector_420.rar.rar` for WebSphere • `ActiveDirConnector_420WL9.rar` for WebLogic	
2	`ActiveDirSchema.jar`	It contains the schema file (`ActiveDir.xml`), which control how Select Identity fields are mapped to Active Directory fields. It also contains properties files, below is an example: `ActiveDirConfig.properties`
3	`cbc_config.zip`	It contains the DDL files to configure the database to block cyclic request.
4	`Password_Installer.zip`	It contains the installation executable for the Password Plug-In.
5	`HP Central AD Agent.zip`	It contains the DLL files, executable, and configuration file for the HP Central AD Agent.

System Requirements

The Active Directory Bidirectional LDAP connector is supported in the following environment:

Table 4 Platform Matrix for Active Directory Bidirectional LDAP Connector

Select Identity Version	Application Server and Operating System	Database
4.0-4.20	The Active Directory Bidirectional LDAP connector is supported on all the platform configurations of Select Identity 4.0-4.20.	

The Active Directory Bidirectional LDAP connector is supported on Microsoft Windows Server 2000 and Microsoft Windows Server 2003 with Service Pack 1.

The Active Directory Bidirectional LDAP connector is internationalized and able to operate with languages that are supported by the Java Unicode specification. If you wish to use the connector on non-English platforms, make sure that the following prerequisites are met:

- The Select Identity server should be configured for internationalization. Refer to the *HP Select Identity Installation and Configuration Guide* for more information.
- The resource should be configured to support local language characters.

Pre-Installation Tasks

To provision users directly to LDAP store, the connector must communicate with the Active Directory resource over a secure channel (LDAPS). To enable a secure communication between the connector and Active Directory, you must perform the following tasks:

- Download CA Certificate to Select Identity Server from Active Directory Server
  - — Download A Certificate
  - — Export the Certificate

  For information about CA certificate generation, see Generating A Root CA Certificate on Active Directory on page 71, and Generating Information for Applying for A New Certificate on page 73.

Before you start installing the connector, you must enable the Secure Socket Layer (SSL) connectivity between Select Identity and the Active Directory Server:

- Configuring SSL Connection Between Select Identity and Active Directory Server
  - — Install Active Directory Certificate on Application Server
    - – WebLogic 8/9 and WebSphere 5
    - – WebSphere 6.1

  In order to enable mutual authentication on Select Identity 4.20, you also need to perform the following tasks:

  - — Configuring for Two-Way (Mutual) Authentication on Select Identity 4.20
    - – Configure for Mutual Authentication

– Rotate Keys

Download CA Certificate to Select Identity Server from Active Directory Server

Download the certificate to the Select Identity server from the Active Directory server by loading the following URL in a browser on the Select Identity server:

http://AD_host/certsrv

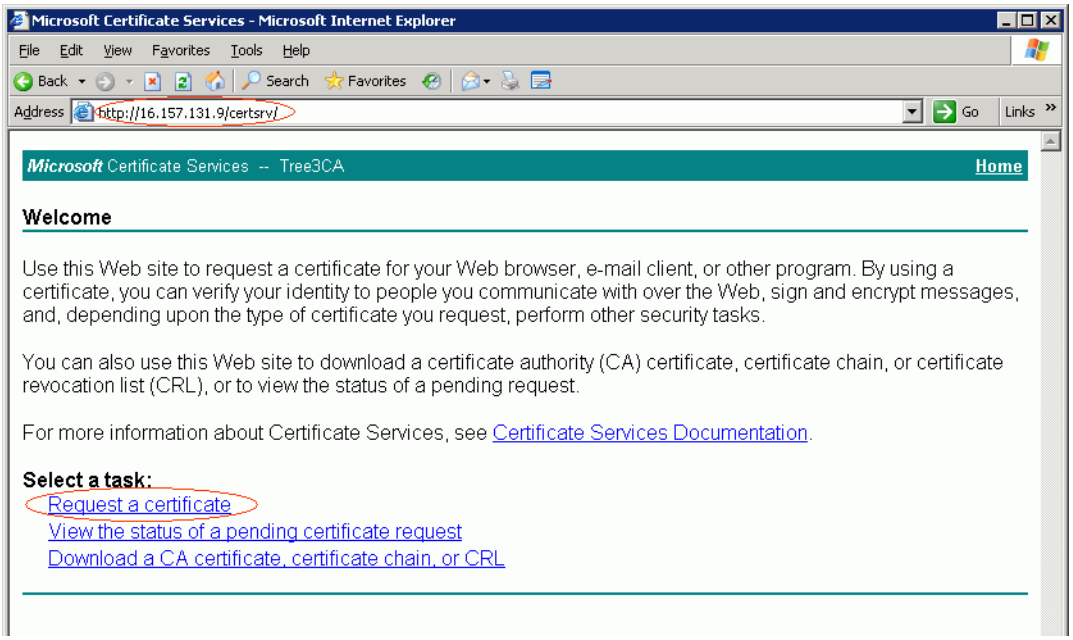
Specify the login credentials for the Active Directory server when prompted. You must download the certificate to the <*Application Server Java Home*>`\jre\lib\security` directory.

You can also copy the certificate to the Select Identity server.

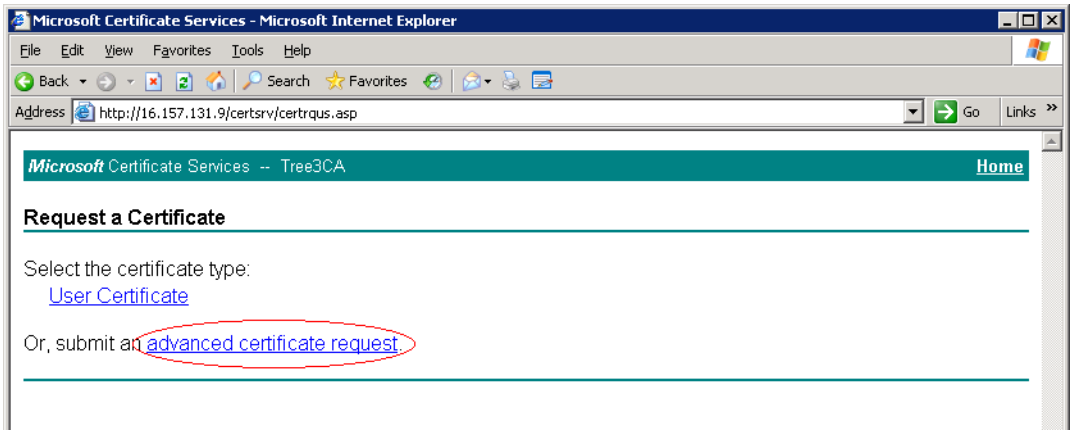
Download A Certificate

1 On your CA server, open Internet Explorer.

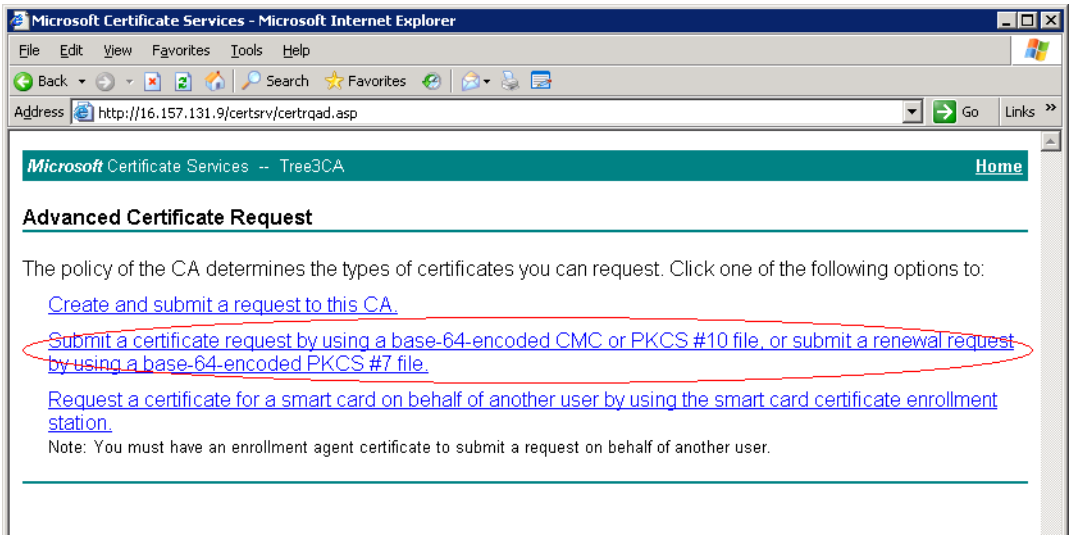
In Address field, enter **`http://localhost/certsrv/`** or **`http://`*`certificate server's IP`*`/certsrv/`**, then click on **Request a certificate** link to open next page.



2 In Request a certificate page, click **advanced certificate request** link.

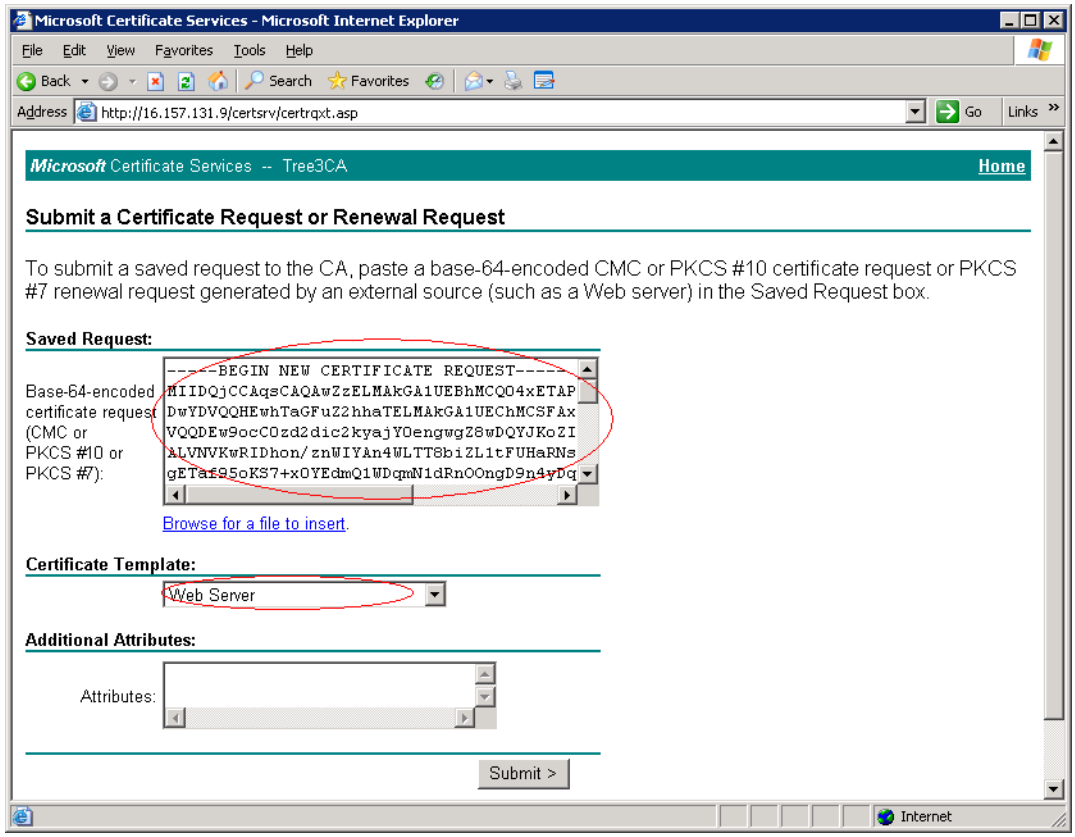


3 Click the second link as shown below:

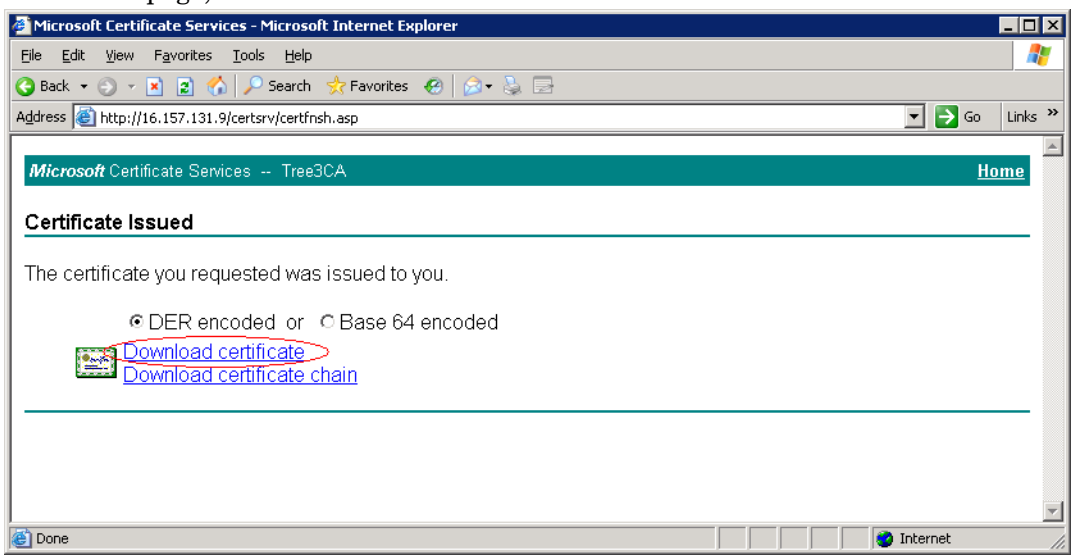


4 Copy request information to the Saved Request field; in Certificate Template filed, select Web Server. Then click **Submit**.

For instructions on how to generate request information, see Generating Information for Applying for A New Certificate on page 73.



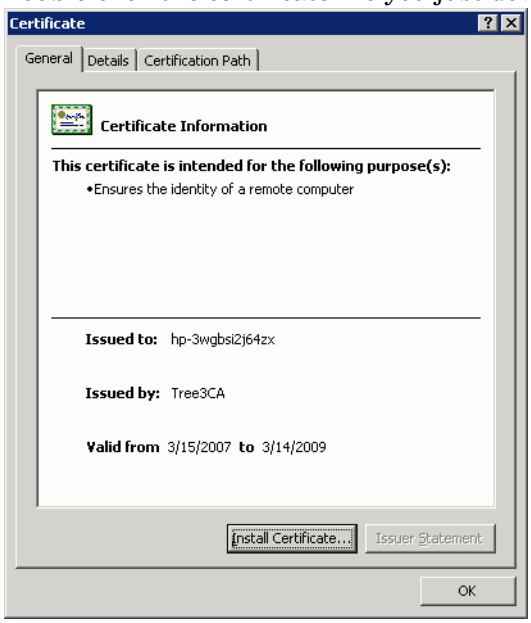
5 In the next page, click **Download certificate** link.



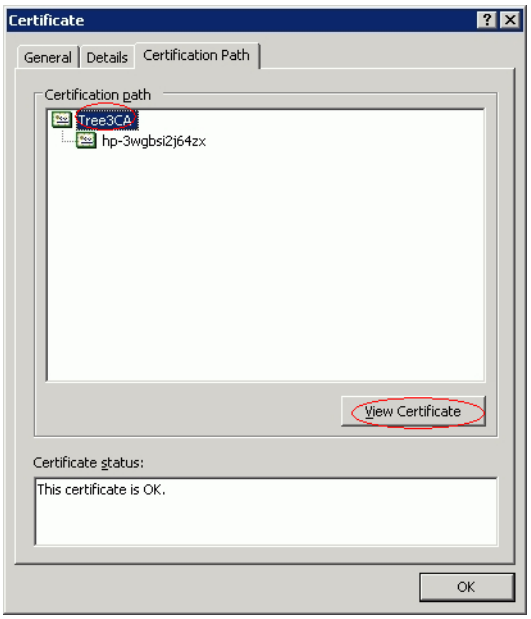
Download the new certificate and save to your local disk.

Export the Certificate

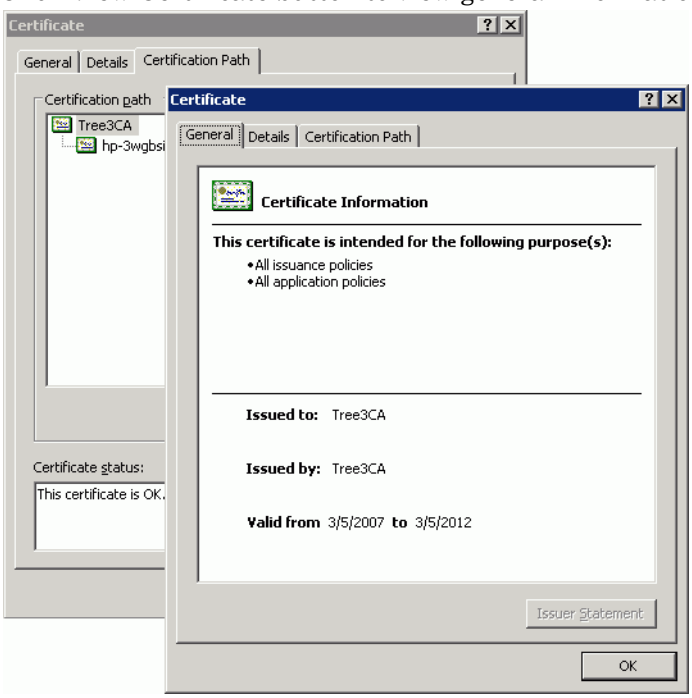
1 Double-click the certificate file you just downloaded to open it.



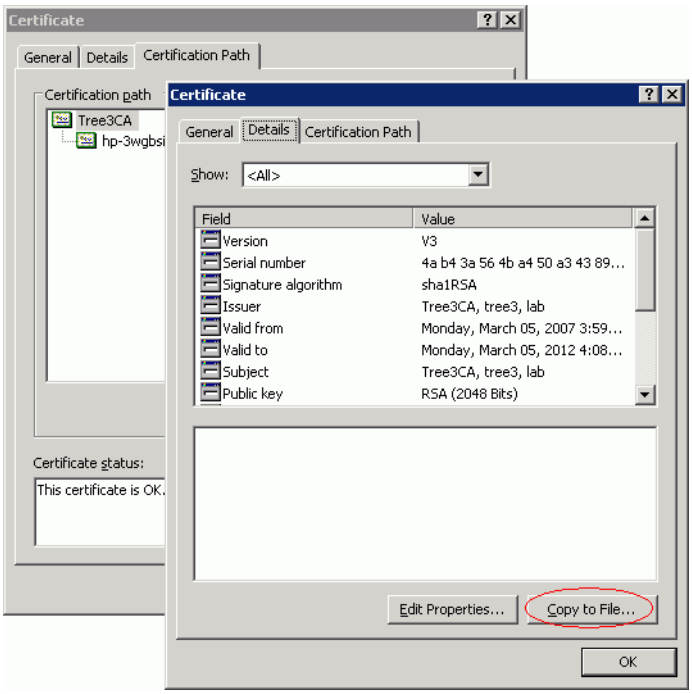
Check certificate path in the Certificate Path tab.



Click View Certificate button to view general information of the certificate.



2 Click Details tab, then click **Copy to File** button.

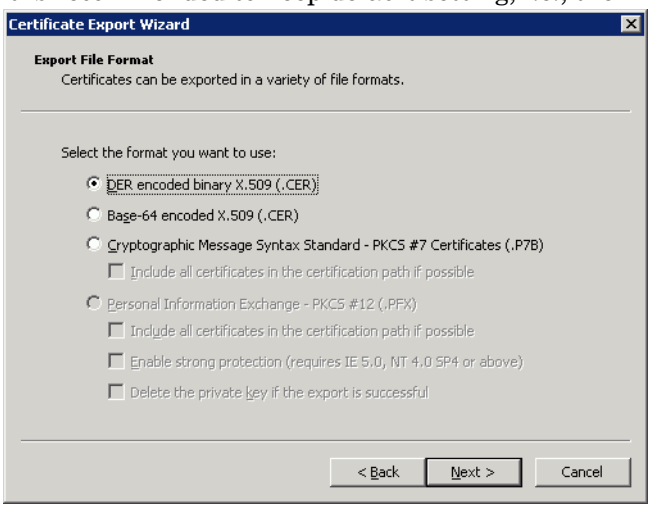


The Certificate Export Wizard opens.

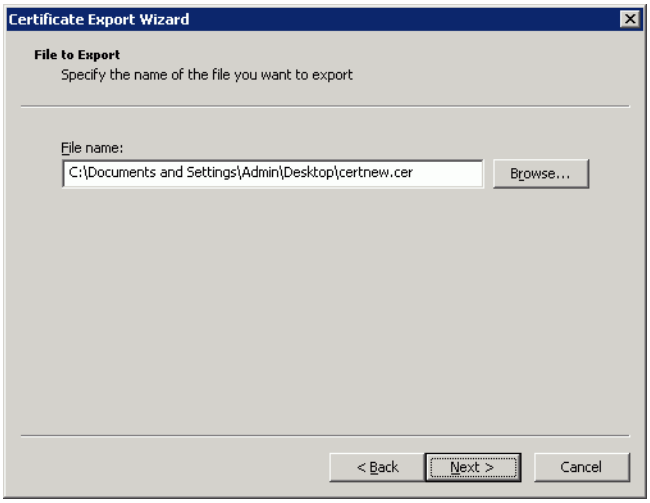
3 Click **Next**.

There are three options available for certificate format, but only first two options work fine.

It is recommended to keep default setting, i.e., the first option.



4 Click **Next**. Specify the name of the file you want to export.



5 Click **Next**.

6 Click **Finish** to export the root certificate.

Configuring SSL Connection Between Select Identity and Active Directory Server

For Select Identity 4.10-4.13, only Active Directory server authentication is supported.

For Select Identity 4.20, Select Identity supports both one-way SSL authentication, in which only the Active Directory server is authenticated, and two-way (mutual) SSL authentication, in which both the Active Directory server and Select Identity are authenticated. For detailed instructions about how to enable one-way or two-way authentication, refer to Configure for Mutual Authentication Support on page 54.

To connect through one-way SSL connection, a server certificate presenting Active Directory resource or a third party should be imported into the Select Identity JDK truststore.

To connect through two-way SSL connection, in addition to importing the Active Directory server certificate or a third party certificate into the Select Identity managed truststore, it is also required to import the certificate presenting Select Identity into the Select Identity managed keystore and Active Directory computer's Trusted Root CA Certificate Store, then map a user to the Select Identity certificate in AD (the user should have the same permissions as the one you created for one-way SSL connection).

- If CRL Validation and Certificate Usage Validation are both disabled, you can choose to use Active Directory certificate or a third party certificate as the server certificate;
- If CRL Validation or Certificate Usage Validation is enabled, only a third party certificate can be used as the server certificate. For detailed instructions, refer to Configure for Mutual Authentication Support on page 54.

Table 5 shows the task matrix for using AD certificate or a third party certificate for one-way/two-way SSL authentication:

Before you start the configuration tasks, make sure that AD SSL connection is enabled.

After you finish the configuration tasks, restart the AD server.

Table 5 Task Matrix

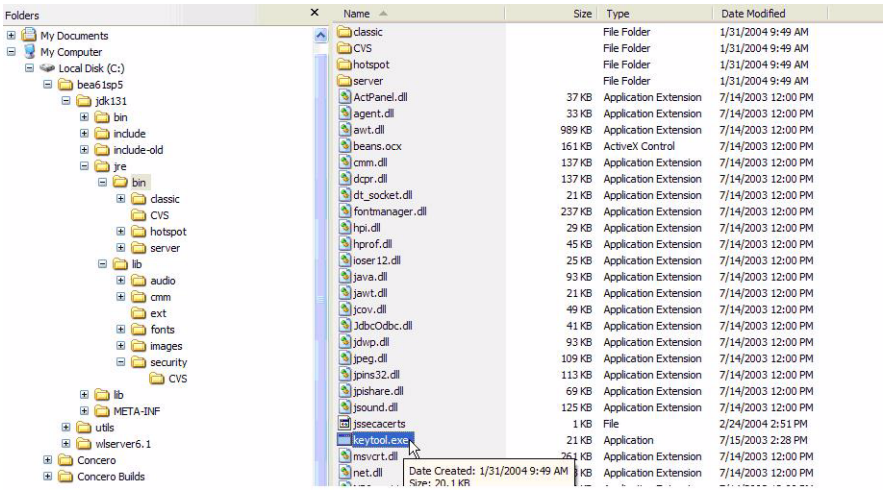
		One-way SSL Authentication	Two-way (mutual) SSL Authentication
Using Active Directory Server Certificate	Select Identity side	• Import the AD root certificate into JDK trust store *For detailed instructions, refer to* Install Active Directory Certificate on Application Server *on page 25.*	• Import the AD root certificate and Select Identity root certificate into Select Identity trust store • Import Select Identity certificate into Select Identity keystore *For detailed instructions, refer to* Configuring for Two-Way (Mutual) Authentication on Select Identity 4.20 *on page 30.*
	Active Directory side		• Import the Select Identity root certificate into AD computer's Trusted Root CA Certificate Store • Map Select Identity certificate to the Admin user *For detailed instructions, refer to* Appendix C.
Using a Third Party Certificate	Select Identity side	• Import the third party root certificate (which is used to sign AD certificate) into JDK trust store *For detailed instructions, refer to* Install Active Directory Certificate on Application Server *on page 25.*	• Import the third party root certificate (which is used to sign AD certificate) and Select Identity root certificate into Select Identity trust store • Import Select Identity certificate into Select Identity keystore *For detailed instructions, refer to* Configuring for Two-Way (Mutual) Authentication on Select Identity 4.20 *on page 30.*
	Active Directory side	• Import the third party certificate signed AD certificate into AD computer's Personal Certificate Store • Import the third party root certificate (which is used to sign AD certificate) into AD computer's Trusted Root CA Certificate Store *For detailed instructions, refer to* Appendix C.	• Import the third party root certificate (which is used to sign AD certificate) and Select Identity root certificate into AD computer's Trusted Root CA Certificate Store • Import the third party signed AD certificate into AD computer's Personal Certificate Store • Map Select Identity certificate to user *For detailed instructions, refer to* Appendix C.

Install Active Directory Certificate on Application Server

WebLogic 8/9 and WebSphere 5

Perform the following steps to install Active Directory certificate on Select Identity:

1. Before installing the Active Directory certificate on application server, verify if `keytool.exe` is available. To verify, go to the Java home of the application server and verify if the `keytool.exe` file is available in <*Application Server Java Home*>`/jre/bin` subdirectory. If Select Identity is installed on Windows, you can locate the file at <*Application Server Java Home*>`/jre/bin` by using Windows explorer.



2. Make sure that the Active Directory certificate file (<*certificate name*>`.cer`) resides in the location <*Application Server Java Home*>`\jre\lib\security` on the Select Identity system.

   ➤ Make sure to copy the certificate to the location <*Application Server Java Home*>`\jre\lib\security` on all application servers for cluster setup purpose.

3. From <*Application Server Java Home*>`jre\bin`, by using command prompt, run the command **`keytool -v -keystore jssecacerts -trustcacerts -import -file ..\lib\security\<certificate name>.cer`** to generate **jssecacerts** file.

   Then, copy the **jssecacerts** file you just generated back to <*Application Server Java Home*>`\jre\lib\security` folder.

4. When prompted for password, enter keystore password (the default password is **`changeit`**).

5. The keytool displays the following message:

```
Owner: CN=QA.hp.com, OU=QA, O="hp", L=abc, ST=xyz, C=mno,
EmailAddress=qa@hp.com
Issuer: CN=QA.hp.com, OU=QA, O="hp", L=abc, ST=xyz, C=mno,
EmailAddress=qa@hp.com
Serial number: 16bab38264ebda84f8011cf35d0ca6a
Valid from: Fri Jan 23 13:42:18 CST 2004 until: Fri Jan 23 13:50:22 CST
2009
Certificate fingerprints:
MD5:  60:72:A9:DD:C4:39:C4:8A:E7:42:56:0B:9E:5D:91:DB
SHA1: 38:D2:7F:33:FE:0A:AC:F3:D3:A0:2C:0F:A9:0C:6A:09:10:B5:EA:66
```

6 If the system displays `Trust this certificate? [no]:`, enter **yes** or **y**. The keytool displays the following message:

```
Certificate was added to keystore
[Saving jssecacerts]
```

7 Copy the new `jssecacerts` file to the *<Application Server Java Home>*`\jre\lib\security` folder.

> Make sure to copy this file, because there is already a `jssecacerts` file in the security folder that needs to be overridden by this one.

8 Restart the application server.

You can add additional certificates by using `alias` flag. For example, after performing the above steps, if you run

keytool -v -keystore jssecacerts -trustcacerts -import -file ..\lib\security*<cert-ADsample.cer>*

you will get an error message:

```
keytool error: java.lang.Exception: Certificate not imported, alias <mykey>
already exists.
```

A listing of the `jssecacerts` shows the `mykey` alias as the default for the just-entered certificate:

```
mykey, Dec 22, 2004, trustedCertEntry,
Certificate fingerprint (MD5):B2:F6:42:F6:0C:88:65:EE:FB:38:3E:31:00:CA:DD:70
```

To get a listing of `jssecacerts`, run the following command:

keytool -list -keystore jssecacerts

To add the additional certificate *`cert-ADsample.cer`*, run the following command:

keytool -v -keystore jssecacerts -trustcacerts -alias hp69trustca -import -file ..\lib\security*cert-ADsample.cer*

The list of jssecacerts now includes:

```
hp69trustca, Dec 22, 2004, trustedCertEntry,
Certificate fingerprint (MD5):60:72:A9:DD:C4:39:C4:8A:E7:42:56:0B:9E:5D:91:DB
```

WebSphere 6.1

Perform the following steps to create keystore file and configure WebShpere 6.1 to use the newly created keystore:

1 Create keystore file:

   a Copy the LDAP certificate file (*<certificate name>*`.cer`) to Select Identity system under *<certificate path>*.

   b Run the command **keytool -v -keystore *<keystore name>* -import -file *<certificate path>*/*<certificate name>*.cer**.

   c When prompted for password, enter your keystore password.

   d The keytool displays a message similar to the following:

```
Owner: CN=QA.hp.com, OU=QA, O="hp", L=abc, ST=efg, C=ab,

EmailAddress=qa@hp.com

Issuer: CN=QA.hp.com, OU=QA, O="hp", L=abc, ST=efg, C=ab,
```

```
EmailAddress=qa@hp.com
Serial number: 16bab38264ebda84f8011cf35d0ca6a
Valid from: Fri Jan 23 13:42:18 CST 2004 until: Fri Jan 23 13:50:22 CST
2009
Certificate fingerprints:
MD5: 60:72:A9:DD:C4:39:C4:8A:E7:42:56:0B:9E:5D:91:DB
SHA1: 38:D2:7F:33:FE:0A:AC:F3:D3:A0:2C:0F:A9:0C:6A:09:10:B5:EA:66
```

e If the system displays `Trust this certificate? [no]:`, enter **`yes`**. The keytool displays the following message:

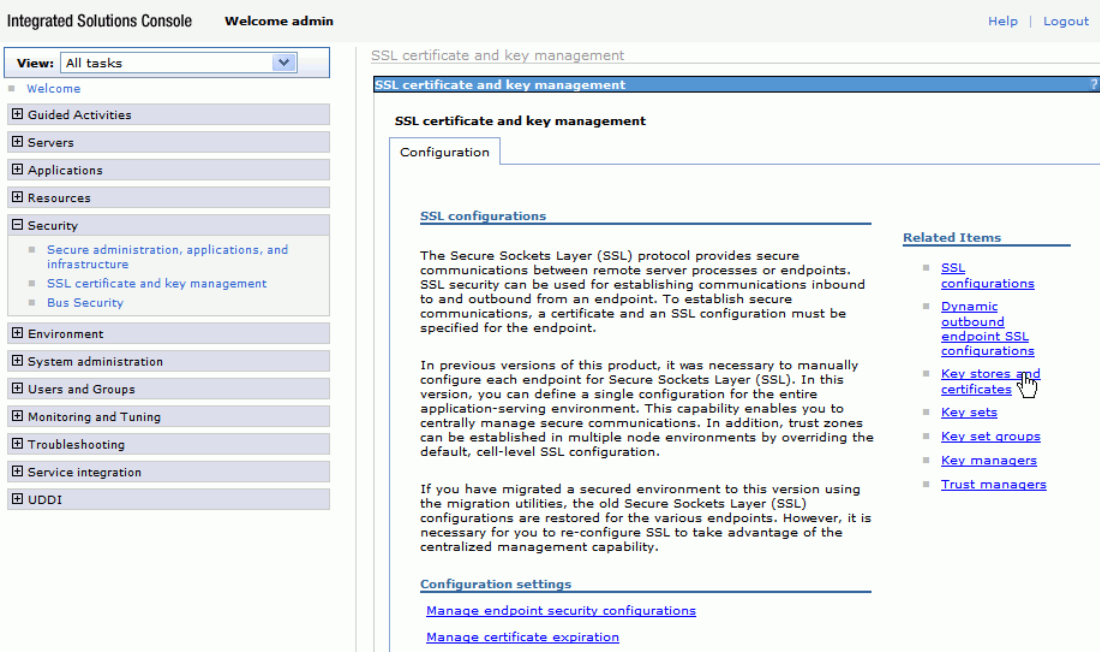
```
Certificate was added to keystore
```

2 Configure WebSphere 6.1 to use the newly created keystore:

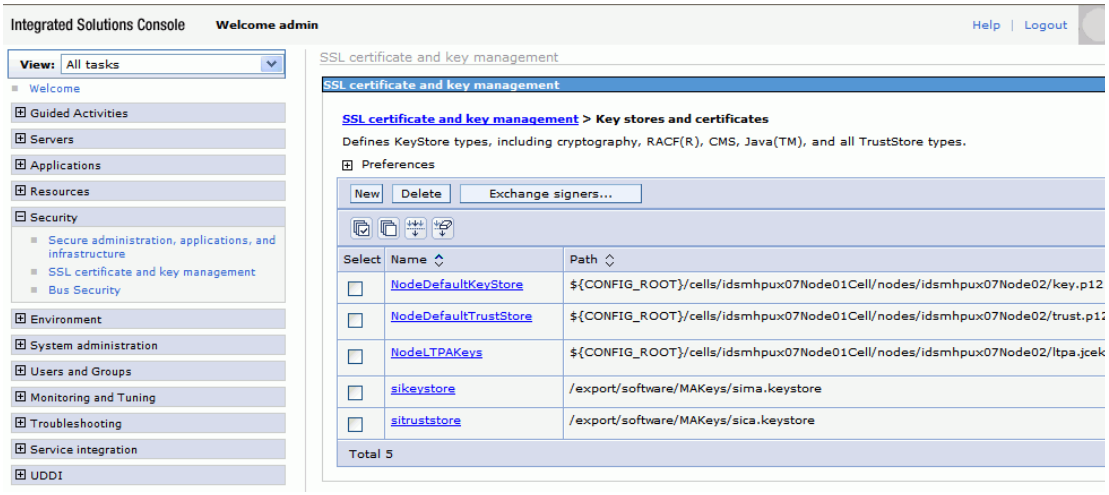
a Logon to WebSphere application server console.

b In the navigation pane, click **Security** → **SSL certificate and key management**. The SSl certificate and key management page displays.

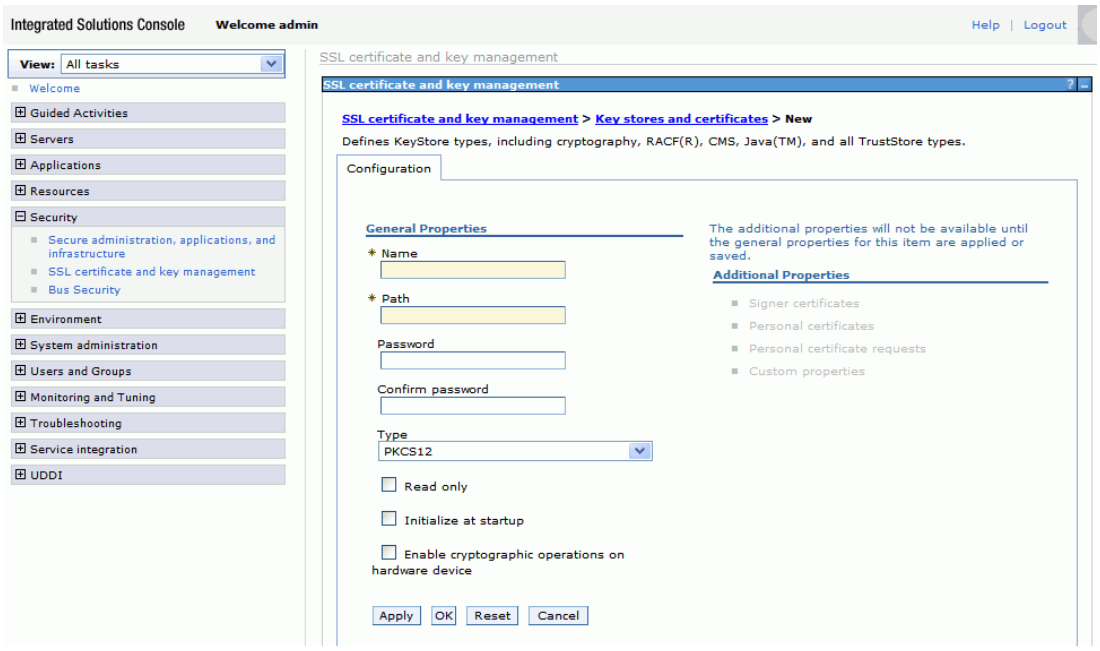
c Under **Related Items** section, click **Key Stores and certificates**. The Key stores and certificates page displays, this is where you can define logical key store that points to the key store file you previously created.



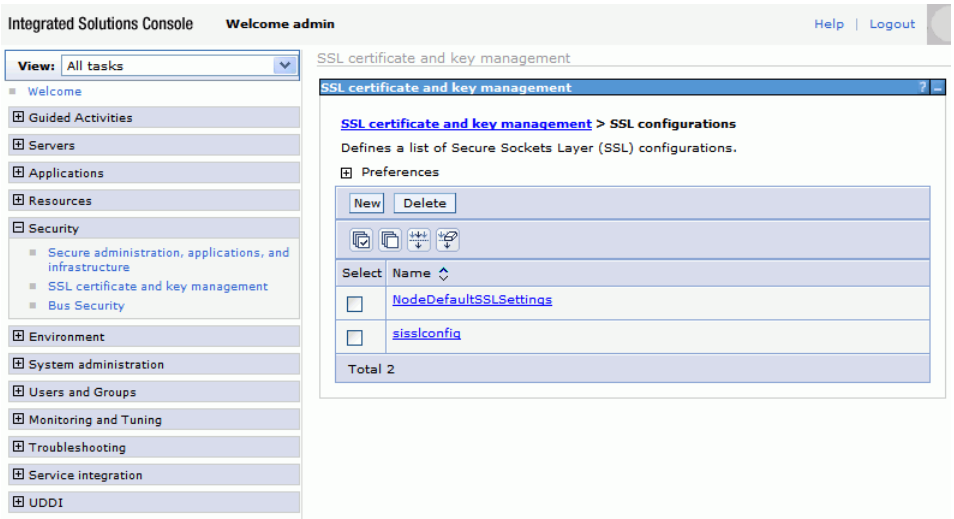
d To create logical trust stores, click **New**.



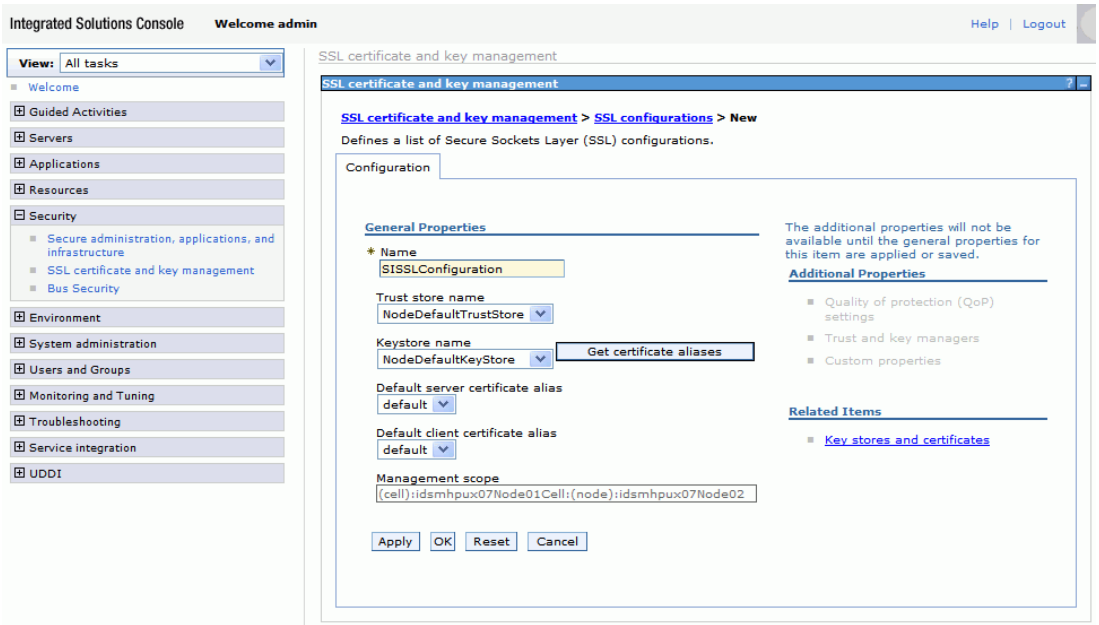
e Input a key store name, key store path (point to the key store file you previously created), password and key store type (should be JKS) for your logical trust store.



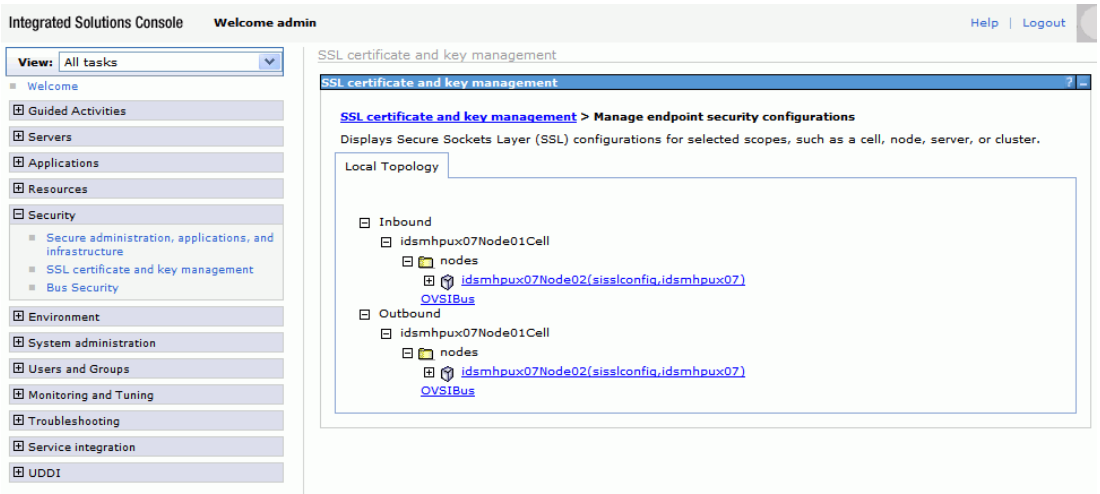
f Go back to SSL certificate and key management page, click **SSL configurations** in **Related Items** section. The SSL configuration page displays.



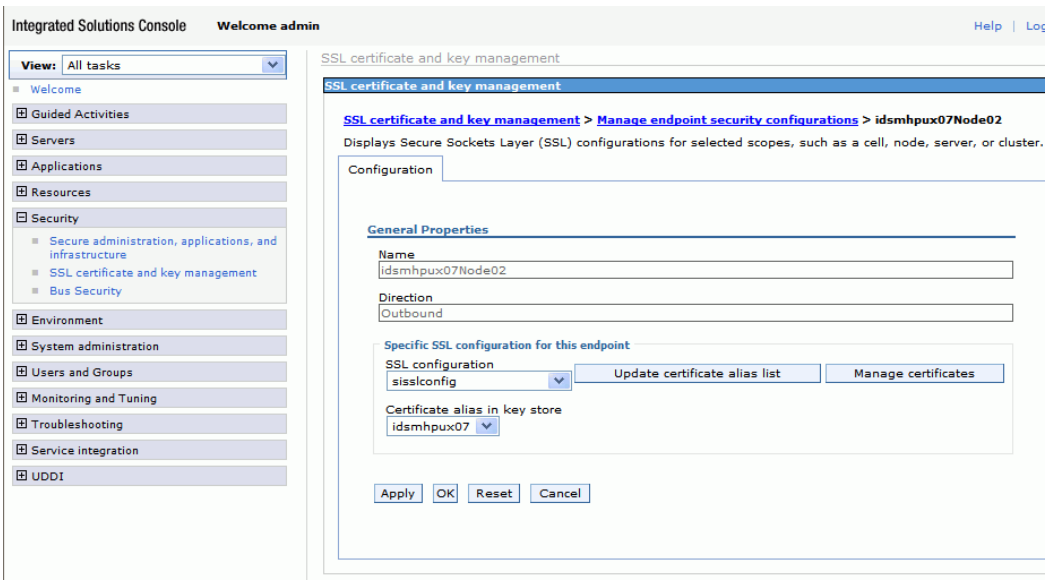
g Click **New.** Define a new SSL configuration that fits your need. Your SSL configuration points to the new logical trust store you defined earlier.



h Go back to SSL certificate and key management page, click **Manage endpoint security configurations** under **Configuration settings** section, then expand **Outbound**.



i Select your SSL configuration and certificate alias.



j Apply your changes and make sure your setting is saved by WebSphere.

Configuring for Two-Way (Mutual) Authentication on Select Identity 4.20

Configure for Mutual Authentication

Perform the following steps to install the Active Directory Bidirectional LDAP certificate:

1 Create and configure Select Identity trust store and properties, if not already created.

a Create the trust store;

b Generate a properties file that is corresponding to the trust store file.

Refer to *Creating the Trust Store* section of *HP Select Identity Installation Guide* for detailed instructions on creating keystore, trust store, and properties.

2 Import certificate representing Active Directory resource to Select Identity trust store:

a Get Active Directory certificate;

b Import the certificate into the trust store file you created in the previous step.

Refer to *Creating the Trust Store* section of *HP Select Identity Installation Guide* for detailed instructions on creating keystore, trust store, and properties.

3 If a resource requires a specific client certificate, you must either generate the client certificate or import the client certificate into the key store:

a Create the key store file;

b Generate the certificate that represents Select Identity server if no certificate available. Or, import the certificate that represents Select Identity server if a certificate already exists.

c Generate the properties file that is corresponding to the keystore.

For more information, refer to *Creating the Key Store and Key Pairs for Mutual Authentication and/or Secure Object Migration* section of *HP Select Identity Installation Guide*.

4 Register the key store and trust store and select the Select Identity client certificate, if not already done.

a Open the security setup tool in Select Identity;

b Register the keystore properties to Select Identity;

c Register the trust store properties to Select Identity;

d Select the certificate representing Select Identity server if needed.

For detailed instructions, refer to *Configure System Security* topic in *HP Select Identity Administration Online Help*.

Rotate Keys

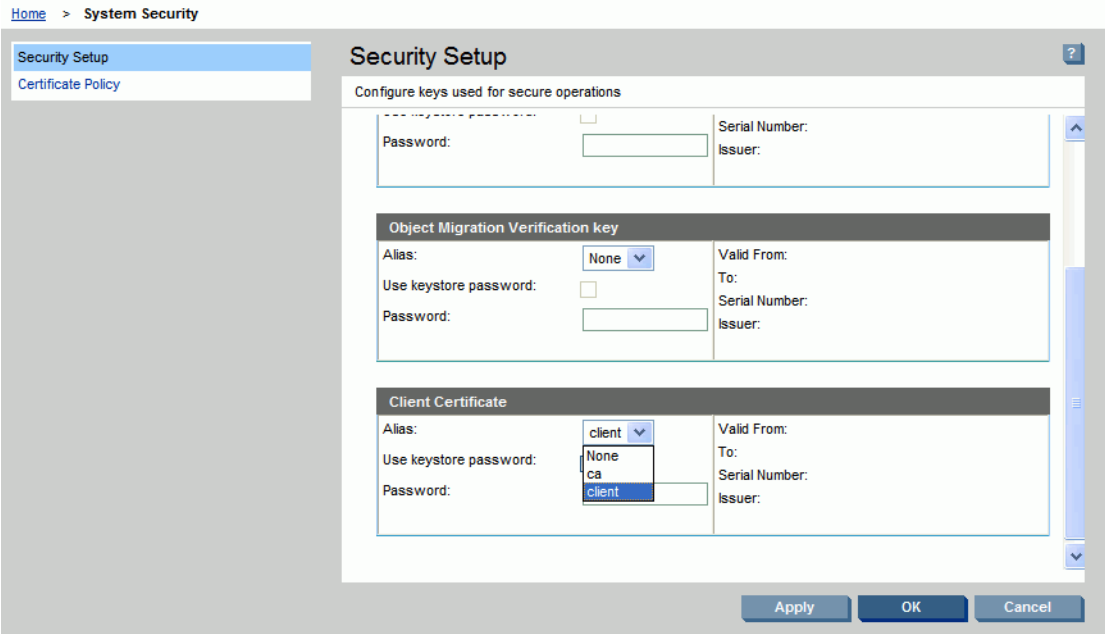
Key rotation is a process that Select Identity can use different keys to connect to a resource. The process is:

1 Generate a new key pair in keystore.

For detailed instructions, refer to *Creating the Mutual Authentication Key store* section of *HP Select Identity Installation Guide*.

2 Change key alias in system security setup:

a From the **Tools** menu, select **System Security** → **Security Setup**. The Security Setup page displays.



b Under Client Certificate section, select the newly generated certificate.

Extracting Contents of the Schema File

The Schema file of the connector contains necessary mapping information to map resource attributes to Select Identity. Extract contents of the `ActiveDirSchema.jar` file to a directory that is in the application server `CLASSPATH`. Refer to the *HP Select Identity Connector Deployment Guide* for detailed instruction to extract contents of the Schema file.

➤ It is recommended to extract the xml and properties file and put them in the schema folder under Select Identity installation directory.

Verifying Configurable Parameters

The properties files, such as `ActiveDirConfig.properties` file, which are present in the `ActiveDirSchema.jar` file, contain the following configurable parameters. These parameters can be changed manually. Before installing the connector, verify the parameter values and change the values if they don't match with the values mentioned below.

➤ In most cases, there is only one properties file present in the `ActiveDirSchema.jar` file, and normally its name is `ActiveDirConfig.properties`. You can customize the file name for your convenience. For example, you can change ***`ActiveDirConfig`***`.properties` to ***`ADConfigNew`***`.properties` so that it corresponds to a specific resource, especially when you have multiple resources. Note that the file extension shall not be changed.

For information on how to add an attribute manually, see Customizing Schema File on page 85.

Non-Customizable Parameters

The following parameters and their descriptions are your information only. It is recommended NOT to change the values for these parameters.

- `entitlement-delimiter=|`

  It contains the string delimiter that is displayed between an entitlement type and its name.

- `modify_replace=false`

  It is a configuration parameter that can be set to true or false. When it is set to false, Active Directory Bidirectional LDAP Connector uses modify/add and modify/delete operations to support multi-valued attribute. When it is set to true, Active Directory Bidirectional LDAP Connector uses modify/replace operation to support multi-valued attribute.

- `attributeValue-delimiter=|`

  It contains the string delimiter that is used to separate attribute values for multi valued attribute.

- `attribute-begins=[[`

  Begin parameter to wrap the special base64 encoded attribute values while sending to connector from Select Identity.

- `attribute-ends=]]`

  End parameter to wrap the special base64 encoded attribute values while sending to connector from Select Identity.

- `dualLink-support=2`

  This specifies whether a Link is a User Link or a Group Link. If it is 1, then it is a User Link. If it is 2, then it is a Group Link.

- `unlink-before-terminate=false`

  If you want to unlink the entitlements while performing a terminate user operation, set this flag to false.

- `null-entitlement-support=true`

  Set this parameters to `true`.

- `entitlement-provisioning=true`

  If this parameter is set to true, the connector will support entitlement provisioning. Otherwise, entitlements will not be provisioned.

- `ldapv3-pageSize=900`

  Number of entries returned from LDAP API when it is queried.

- `number-of-retries=3`

  Number of retry times of failover.

- `retry-delay=1`

  Retry interval (in seconds).

Customizable Parameters

The following parameters are customizable. You can change the *italic* parts of parameter values below to fit your needs:

> It is NOT recommended to make any changes after you have put the system into production for some time.

- `PSSync_ATTRIBUTE=`*`description`*

  This Active Directory attribute is used by Password Plug-in to temporarily store user encrypted password. This attribute name is saved on both Select Identity AD Connector properties file and Password Plug-In properties file. For more information on configuration of agent ini file (`ADProperties.ini`), see step 12 on page 47.

  If the password plug-in is not installed, the value can be empty (for example, you can configure it like this: `PSSync_ATTRIBUTE=`).

- `OVSI.ADConnector.groupid.attribute=`

  This specifies display name of OVSI AD Connector group in Select Identity graphical user interface. There are four values available for this parameter:

  — *`dotFormat`* – the default format for group name will be displayed. It will use "`.`" as separator to show the distinguishedName of the group. For example, if the group's distinguishedName is "cn=group1,OU=Test,DC=root,DC=sicf" in AD, it will show "Group|group1.Test.root.sicf";

  — *`cn`* – the common name of the group will be displayed. The common name must be **unique** in the forest, as in multi-domain, the cn can be duplicated in different domains. Therefore, if you want to use cn as the group's display name, **make sure that the cn must be unique in the forest**. This is a limitation for using cn as the display name for group.

  — *`distinguishedName`* – the distinguished name of the group will be displayed;

  — *`description`* – the description of the group will be displayed. The description must be **unique** in the forest, and the description supports maximum 100 characters; If the description is empty, the parameter will take its *`cn`* as the group display name. This is a limitation for using description as the display name for group.

  > It is recommended to use `dotFormat` or `distinguishedName` value.

The following five parameters are for moving user across domain function:

- `OVSI.Command.Message.Request.Attribute=info`

  It specifies the Active Directory attribute to temporarily store request info for moving user across domain.

- `OVSI.Command.Message.Response.Attribute=info`

  It specifies the Active Directory attribute to temporarily store response info for moving user across domain.

- `OVSI.Command.Message.Delimeter=#####`

  Used in request and response info to separate parameters for moving user across domain.

  > Make sure that the above three attributes have the same attribute values as those in `PasswordAgent-config.xml` (present in `System32` directory of the machine on which support for moving user across domain is enabled) as shown below:

```
<?xml version="1.0" encoding="utf-8" ?>
- <PasswordAgent-config>
  - <constants-config>
      <constant name="request" attribute="info" />
      <constant name="response" attribute="info" />
      <constant name="delimiter" value="#####" />
    </constants-config>
  - <action-mappings>
      <action message="moveUserAcrossDomains" assembly="HP.AD.WNF.MoveUser"
        className="HP.AD.WNF.MoveUser.MoveUserAction" />
    </action-mappings>
  </PasswordAgent-config>
```

- `OVSI.Command.Message.DeleteTransientUser=true`

  It specifies whether to delete transient user in Active Directory when move user across domain is finished.

- `OVSI.Command.Message.Retrieve.Intervals=10`

  Retry interval (in seconds).

- `OVSI.Command.Message.Retrieve.Times=8`

  Number of retries.

- `# AD forest configuration`

  `# ------------------------`

  `OVSI.ADConnector.gc.count=`***1***

  `OVSI.ADConnector.gc.0=`*rootdc1.root.sicf*

  `OVSI.ADConnector.gc.0.port=`*3268*

  `OVSI.ADConnector.gc.0.domain=`*dc=root,dc=sicf*

  `OVSI.ADConnector.domain.count=`***3***

  `# Domain 1`

  `OVSI.ADConnector.domain.0=`*dc=root,dc=sicf*

  `OVSI.ADConnector.domain.0.userSuffix=`*ou=selectidentity,ou=openview*

  `OVSI.ADConnector.domain.0.groupSuffix=`*ou=selectidentity,ou=openview*

  `OVSI.ADConnector.domain.0.transientUserSuffix=ou=`*transientuserSuffix*

  `OVSI.ADConnector.domain.0.dc.count=`*2*

  `OVSI.ADConnector.domain.0.dc.0=`*rootdc1.root.sicf*

  `OVSI.ADConnector.domain.0.dc.0.port=`*636*

  `OVSI.ADConnector.domain.0.dc.1=`*rootdc2.root.sicf*

  `OVSI.ADConnector.domain.0.dc.1.port=`*636*

  `# Domain 2`

  `OVSI.ADConnector.domain.1=`*dc=child1,dc=root,dc=sicf*

```
OVSI.ADConnector.domain.1.userSuffix=ou=selectidentity,ou=openview
OVSI.ADConnector.domain.1.groupSuffix=ou=selectidentity,ou=openview
OVSI.ADConnector.domain.1.transientUserSuffix=ou=transientuserSuffix
OVSI.ADConnector.domain.1.dc.count=1
OVSI.ADConnector.domain.1.dc.0=child1dc1.child1.root.sicf
OVSI.ADConnector.domain.1.dc.0.port=636

# Domain 3
OVSI.ADConnector.domain.2=dc=child2,dc=root,dc=sicf
OVSI.ADConnector.domain.2.userSuffix=ou=selectidentity,ou=openview
OVSI.ADConnector.domain.2.groupSuffix=ou=selectidentity,ou=openview
OVSI.ADConnector.domain.2.transientUserSuffix=ou=transientuserSuffix
OVSI.ADConnector.domain.2.dc.count=1
OVSI.ADConnector.domain.2.dc.0=child2dc1.child2.root.sicf
OVSI.ADConnector.domain.2.dc.0.port=636
```

Below are explanations to the above properties:

1) `OVSI.ADConnector.gc.count=1`

2) `OVSI.ADConnector.gc.0=rootdc1.root.sicf`

3) `OVSI.ADConnector.gc.0.port=3269`

4) `OVSI.ADConnector.gc.0.domain=dc=root,dc=sicf`

5) `OVSI.ADConnector.domain.count=3`

These five lines are AD forest configuration information:

— 1) The **`OVSI.ADConnector.gc.count`** property determines the number of global catalogs in a forest. In this instance, there is only one global catalog in the forest.

  If **`OVSI.ADConnector.gc.count`** property value is 2, there will be another three lines indicating full name and port number of the machine for the second global catalog and domain name respectively.

— 2) The **`OVSI.ADConnector.gc.0`**`=rootDC1.root.sicf` property indicates that the full name of the machine where the global catalog resides is `rootDC1.root.sicf`;

— 3) The **`OVSI.ADConnector.gc.0.port`**`=3268` property indicates that the port number of the machine is `3268`.

  If one-way authentication is enabled, the port of global catalog should be set to `3268`. If two-way authentication is enabled, the port of global catalog should be set to **`3269`**.

— 4) The **`OVSI.ADConnector.gc.0.domain`**`=DC=root,DC=sicf` property indicates that the domain name is `DC=root,DC=sicf`.

— 5) The **`OVSI.ADConnector.domain.count`** property determines the number of domains in a forest, and the property value varies with your environment. In this instance, the property value is 3, meaning that there are three domains in the environment.

6) `# Domain 1`

7) `OVSI.ADConnector.domain.0=`*`dc=root,dc=sicf`*

8) `OVSI.ADConnector.domain.0.userSuffix=`*`ou=selectidentity,ou=openview`*

`OVSI.ADConnector.domain.0.groupSuffix=`*`ou=selectidentity,ou=openview`*

9) `OVSI.ADConnector.domain.0.transientUserSuffix=`*`ou=transientuserSuffix`*

10) `OVSI.ADConnector.domain.0.dc.count=`*`2`*

11) `OVSI.ADConnector.domain.0.dc.0=`*`rootdc1.root.sicf`*

`...`

The code lines following **`OVSI.ADConnector.domain.count`** property are domain-specific properties information:

For Domain 1,

— 7) the `OVSI.ADConnector.domain.0=`*`dc=root,dc=sicf`* property indicates domain name is `dc=root,dc=sicf`;

— 8) the `OVSI.ADConnector.domain.0.userSuffix` property and `OVSI.ADConnector.domain.0.groupSuffix` property indicate user suffix and group suffix in the domain respectively;

`UserSuffix` is the top user location that connector can provision user and detect user changes. If `UserSuffix` is set to empty, that allows the connector to manage all users in the domain. For example, if there is parent "*ou=openview*" and you want the connector to only manage users in that branch, you can set "*ou=openview*" in the property file. If the user attribute (`UserSuffix` on Select Identity) is set to "ou=ca, ou=openview", the user will be provisioned to the child "ou=ca". (Make sure that the child OU already exists in the domain controller.)

`GroupSuffix` is the top group location that connector can retrieve groups as user entitlement or detect group member changes. It is an known limitation in this release that only one group location can be specified. Also, it cannot be set to empty.

— 9) the `OVSI.ADConnector.domain.0.transientUserSuffix=` property indicate transient user suffix in the domain. When move user across domain, the connector automatically creates a transient user under the `transientUserSuffix` OU. **You only need to make sure that this OU exists on the AD server.**

— 10) the `OVSI.ADConnector.domain.0.dc.count` property indicates the number of domain controllers in the domain; and

— 11) the `OVSI.ADConnector.domain.0.dc.0` property indicates the full name of the machine where the DC resides.

The rest may be deduced by this analogy.

For forward or reverse provisioning, only groups within the scope specified by **`groupSuffix`** will be displayed on Select Identity server.

Installing the Connector RAR

To install the RAR file of the connector (such as `ActiveDirConnector_420.rar`) on the Select Identity server, you must copy the file to a local subdirectory on the Select Identity server, and then deploy on the application server. Refer to *Chapter 4 of HP Select Identity Connector Deployment Guide* for detailed information about deploying a RAR file on an application server.

While deploying the RAR on WebSphere, enter the JNDI Pool Name as **`eis/ActiveDirConnector`**.

Configuring the Database on Select Identity System to Block Cyclic Request

The Active Directory Bidirectional LDAP connector supports both forward provisioning and change detection. When a forward operation is performed on the resource, the next polling cycle of the connector may detect the operation as if it was performed directly on the Active Directory system. This is called cyclic request. To block any cyclic request, you must configure the database of Select Identity.

Perform the following steps to block cyclic request:

1. On the Select Identity database, execute the DDL file (`mssql_cbc_ddl.sql` for Microsoft SQL Server database or `Oracle_cbc_ddl.sql` for Oracle database), which are available in `cbc_config.zip`.
2. Modify `ActiveDirConfig.Properties` file.

   Set the `CBCDatasource — JNDIName` and `CBCDatasource — Repository` parameters as below. The two parameters are stored in `ActiveDirConfig.Properties` file.

   ```
   CBCDatasource — JNDIName=jdbc/TruAccess
   CBCDatasource — Repository=<database type>
   ```

   where `<database type>` is Select Identity's database (`Oracle` for Oracle database and `mssql` for Microsoft SQL Server database).

   Use Select Identity's connection pool and JDBC data source to read/write the database.

Each time when you finish creating a resource, make sure to execute the following script in the database to add corresponding entries into `ovsi_bidirldap_lcln` table. The number of entries is determined by the number of domain controllers in the entire forest.

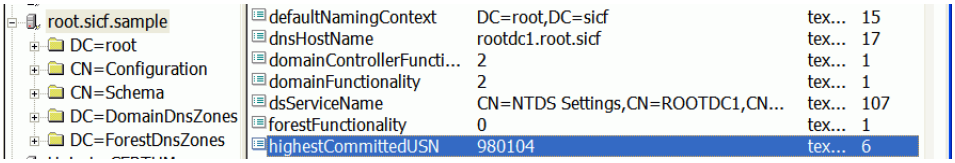
```
insert into ovsi_bidirldap_lcln values('rootDC3.root.sicf','330612','ELDAPADsample')
```

in this instance,

— `'rootDC3.root.sicf'` is the full qualified domain name of the domain controller that performs reconciliation.

— `'330612'` is the last change log number of each domain controller.

To get the last change log number on the Active Directory server, you can use a LDAP browser to retrieve the value of parameter `highestCommittedUSN`, as shown in an example below:



— `'ELDAPADsample'` is the resource name that is created on Select Identity server.

You can also find this script in `cbc_config.zip` package, with the name of `config.sql`.

This SQL statement only applies when there is only one domain controller in a domain that is configured in the configuration file.

4 Installing Agent

This chapter gives an overview of agent for Active Directory Bidirectional LDAP connector. At the end of the chapter, you will be able to know about:

- The role of an agent.
- The procedure to install the agent.

About Agent

The Active Directory Bidirectional LDAP connector is packaged with an agent module—Password Plug-In. The Password Plug-In detects any change in password on the Active Directory system.

Installing Password Plug-In

Make sure to install the Password Plug-In on the Active Directory server (global category) by using the agent installation wizard.

The Password Plug-In detects any change in password on the Active Directory system in order to perform password reconciliation.

If you selected **Support move user across domain** during installation, then the Password Plug-In supports moving user across domain.

Currently the agent has separate versions avaiable for both 32bit and 64bit AD server.

➤ In an Active Directory multi-domain forest environment, run HP Central AD Agent setup utility to distribute the Password Plug-In onto all Domain Controller servers.

When installing HP Central AD Agent, make sure to install it on the same machine where Password Plug-In is installed.

Preparation

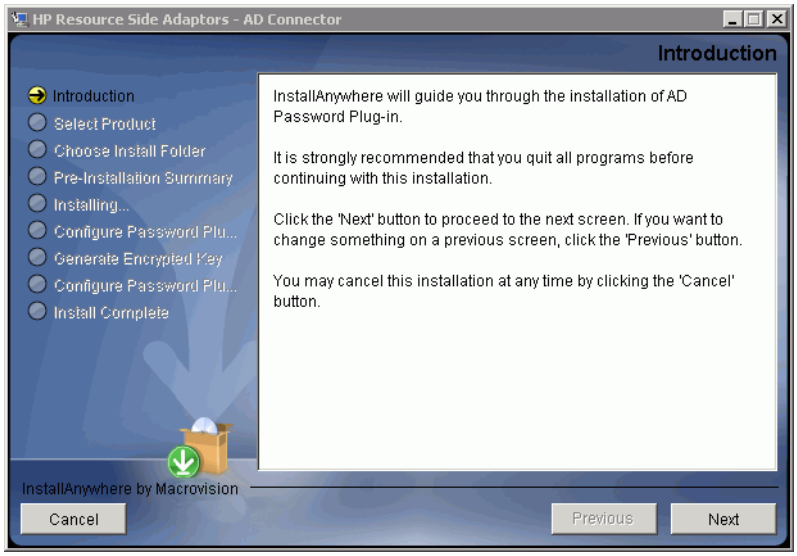
Before you start the installer, perform the step below:

Extract the contents of the file `Password_Installer.zip` to a local directory (*<Installer Dir>)* on the Active Directory system. The automatic folder installer program `setup.exe` is stored in *<Installer Dir>*`\Disk1\InstData\NoVM` directory.

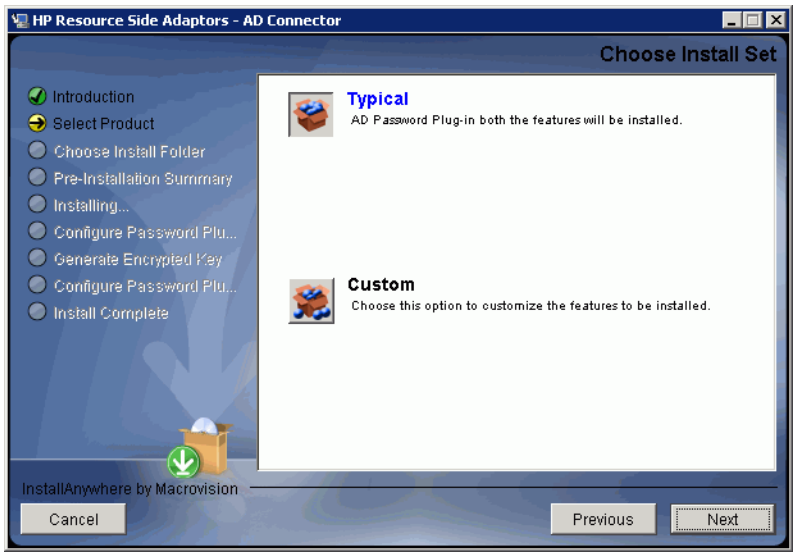
Installation Procedure

Perform the following steps to install password plug-in with the help of the wizard:.

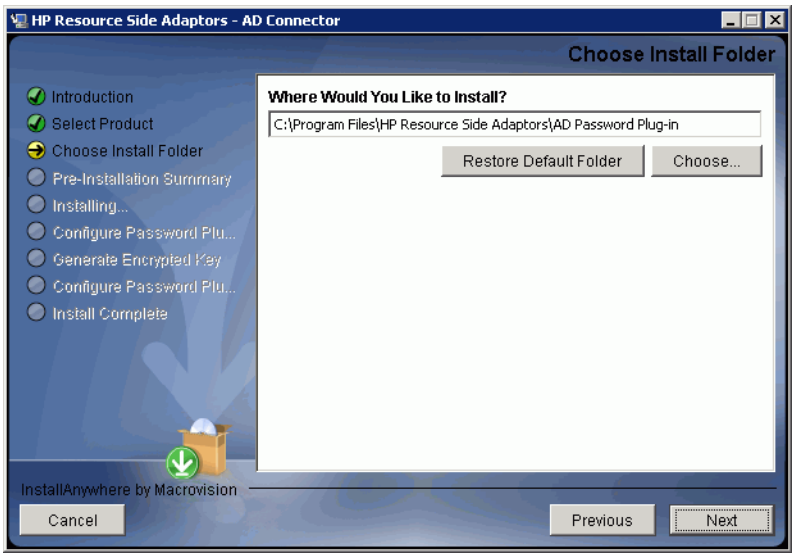
1 Run `setup.exe`, which is located in *<Installer Dir>*`\Disk1\InstData\NoVM` directory at resource system. The installation wizard appears.



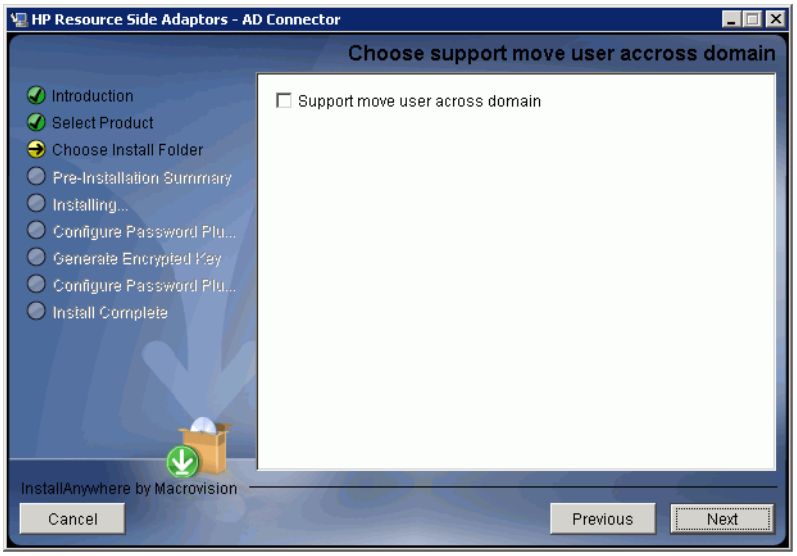
2 Click **Next** to begin installation. Choose Install Set screen appears.



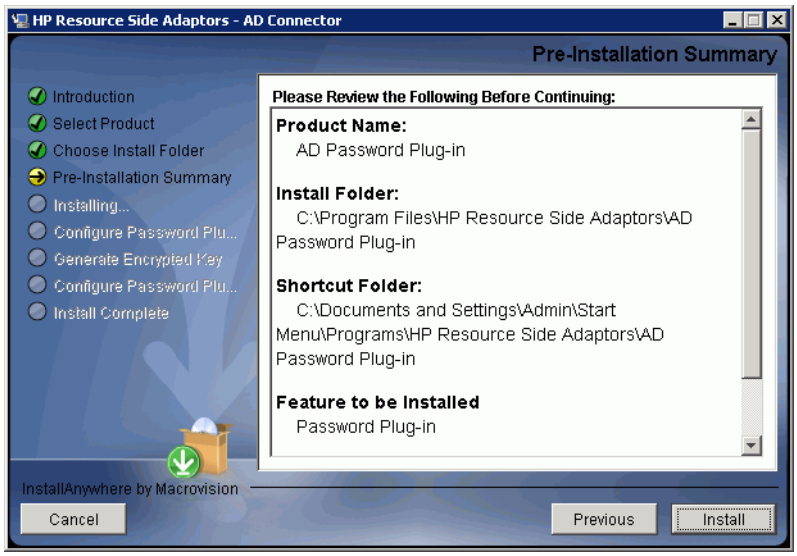
3 Choose Typical install set, and then click **Next**. Choose Install Folder screen appears.



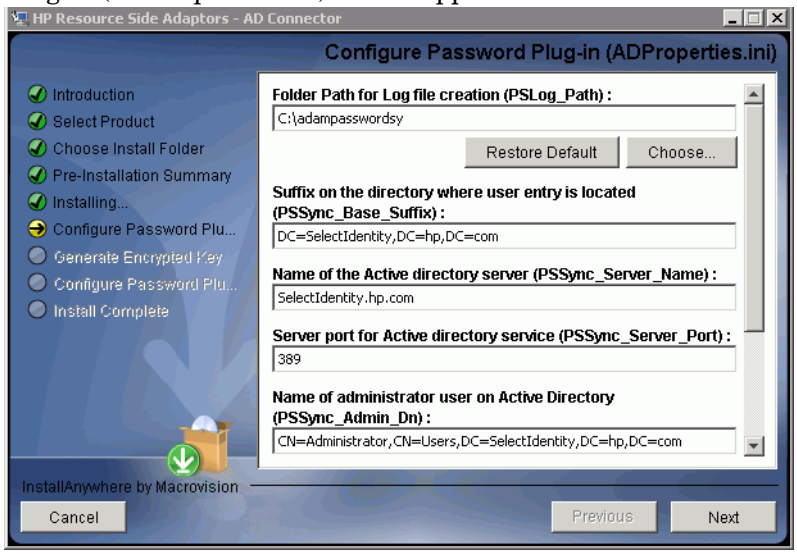
4 If you want to provide support for moving user across domains, select **Support move user across domain**; otherwise, leave it empty.

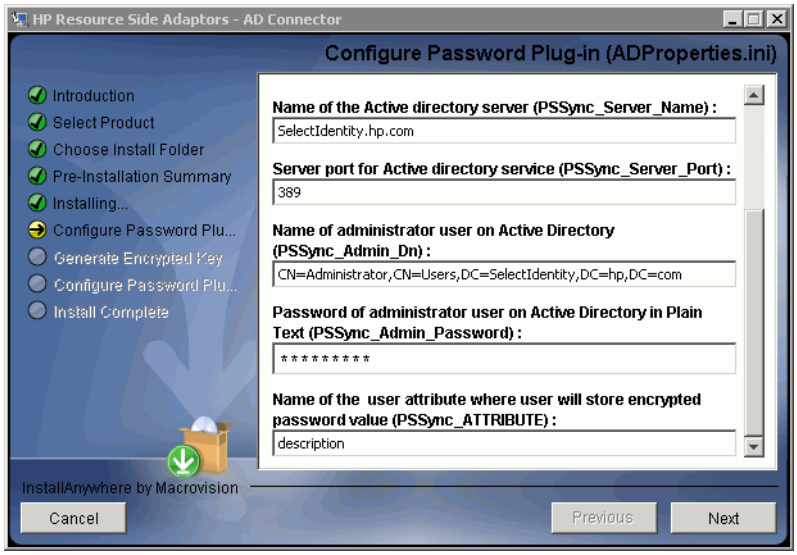


5 Specify the location for password plug-in, and then click **Next**. Pre-Installation Summary screen appears.



6 Review the summary and click **Install** to begin installation. The Configure Password Plug-in (ADProperties.ini) screen appears.



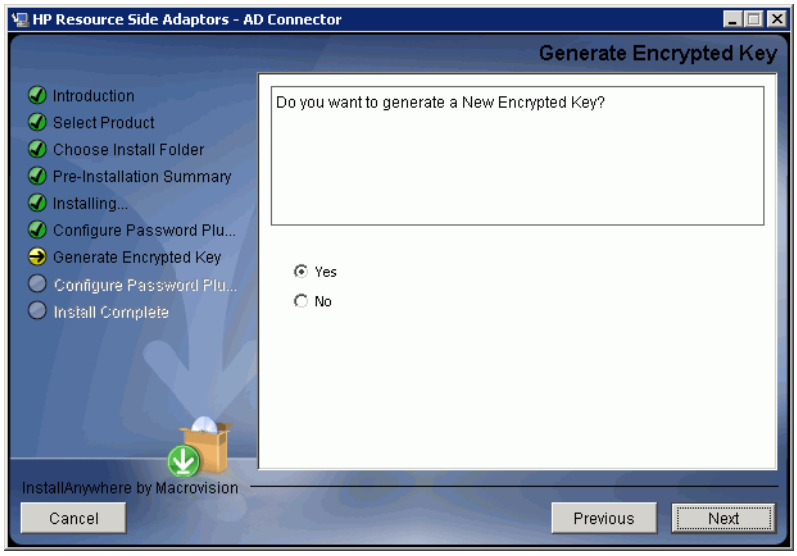


In the text fields, you must enter the following parameters.

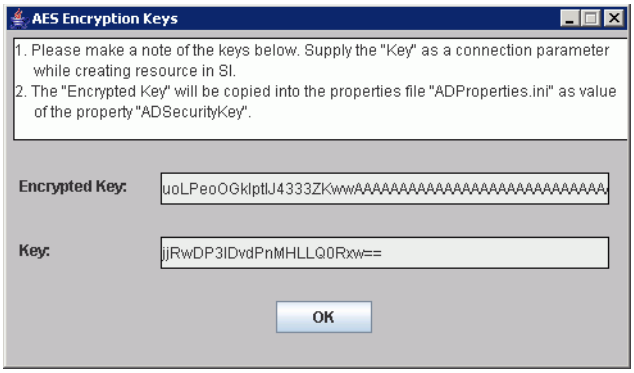
- `PSLog_Path`: The folder name (not filename) under which a log file is created. Mention an existing location on Active Directory server in this field, or create a new folder in Active Directory server and enter the path of the newly created folder.
- `PSSync_Base_Suffix`: This is the base suffix on Active Directory where user entries are located. (For example, DC=SelectIdentity,DC=hp,DC=com)
- `PSSync_Server_Name`: Name of the Active directory server (For example, SelectIdentity.hp.com)
- `PSSync_Server_Port`: Server port for Active directory service (For example 389)
- `PSSync_Admin_Dn`: Name of administrator user on Active Directory (For example, CN=Administrators,CN=Users,DC=SelectIdentity,DC=hp,DC=com)
- `PSSync_Admin_Password`: Password of administrator user on Active Directory in encrypted format.
- `PSSync_ATTRIBUTE`: Name of the user attribute where user will store encrypted password value in the Active Directory. The field which are mentioned should have the capacity of holding more than 180 characters. Otherwise AD will not be able to hold the encrypted password. For example, `description` attribute in Active Directory.

This is a sensitive attribute containing user's encrypted password. It is highly recommended to choose an attribute that is not used by any application and is not easily visible or available. Extending the Active Directory schema for this additional attribute is a good way to make this attribute obscure.

7 Click **Next**. Generate Encrypted Key screen appears.

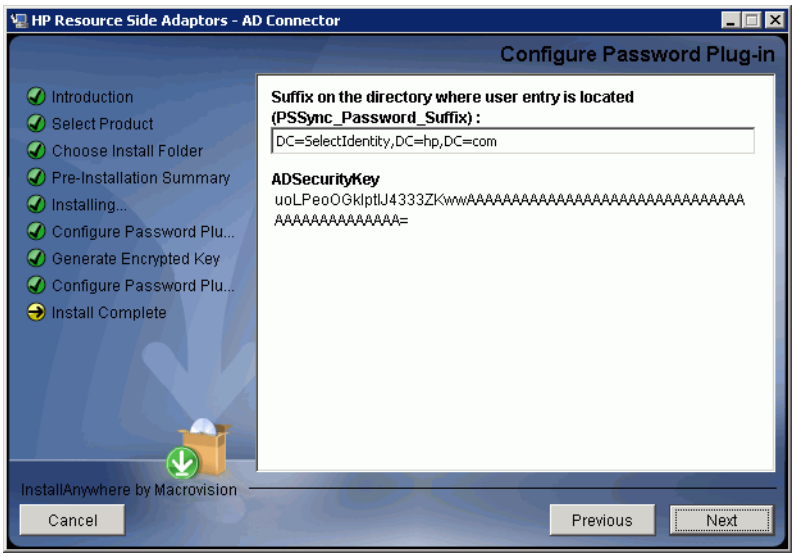


8 Check the Yes radio button and click **Next**. AES Encryption Keys popup appears.



- Note down the value in Key field and save the Key. You must supply this value against Encryption Key field while entering resource access information parameters in Select Identity.
- It is NOT recommended to change the key. If you DO need to change it, make sure to reset all users password.

9 Configure Password Plug-in screen appears.



Suffix on the directory where user entry is located (PSSync_Password_Suffix) :
DC=SelectIdentity,DC=hp,DC=com

ADSecurityKey
uoLPeoOGklptlJ4333ZKwwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA=

10 Enter the `PSSYnc_Password_Suffix` as suffix on the directory where user entry is located. (For example, DC=SelectIdentity,DC=hp,DC=com), and then click **Next**.

11 After the installation is complete, click **Done**.

12 The agent records information about the password plug-in operation in the log file. You can filter this information by setting the `PSLog_Level` attribute in the `ADProperties.ini` file. Perform the following steps to set this attribute:

a Open the `ADProperties.ini` file from the location `C:\WINDOWS\system32`.

b Set the `PSLog_Level` attribute to 0, 1, 2, or 3.

- Set `PSLog_Level` to `0` to record only basic information.
- Set `PSLog_Level` to `1` to record intermediate level information.
- Set `PSLog_Level` to `2` to record advanced level information.
- Set `PSLog_Level` to `3` to record developer level information.

13 Restart the machine after installation. And, remember to BACKUP the `ADProperties.ini` file.

`Description` is the default Active Directory attribute used by Password Plug-In to store encrypted password.

If you want to use a different attribute to store encrypted password, perform the steps below:

- *On Password Plug-In side:*

  Modify Password Plug-In properties file (`ADProperties.ini`): replace "`description`" in "`PSSync_ATTRIBUTE=`*`description`*" with another attribute name.

- *On Select Identity server side:*

  Stop application server, and modify `ActiveDirConfig.properties` in `ActiveDirSchema.jar`: replace `description` in `PSSync_ATTRIBUTE=`*`description`* with another attribute name; then start the application server again.

Distributing Password Plug-In

You can distribute Password Plug-In onto every domain controller in the forest by running HP Central AD Agent setup utility.

Preparations

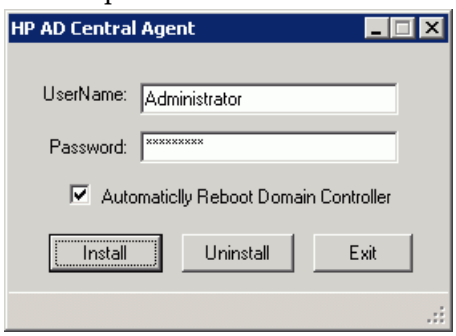
1. Download and install Microsoft .Net framework 2.0, then add the path where `RegAsm.exe` file is located (for example, `C:\WINDOWS\microsoft.net\Framework64\v2.0.50727`) into system variable `Path`.
2. Check that Password Plug-In is installed successfully by installer wizard by verifying the existence of the following four files in `%SystemRoot%\system32` directory:

   `ADProperties.ini`
   `ADPassfilt.dll`
   `libeay32.dll`
   `libssl32.dll` (for 32bit) / `ssleay32.dll` (for 64bit)
3. Extract the contents of the file `HP_Central_AD_Agent.zip` to a local directory (*<Installer Dir>)* on the same AD domain controller server. The `HP Central AD Agent Setup.exe` is stored in *<Installer Dir>*`\HP_Central_AD_Agent` directory.
4. Make sure that the credential used to login on the domain controller has the permission on every domain controller in the forest to execute the following tasks:
   - Access and write permission to `%systemroot%\system32` directory on remote computer
   - Write and modify permission to Registry on remote computer

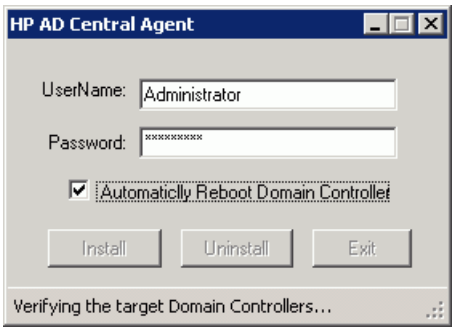
Installation Procedure

Perform the following steps to run HP Central AD agent:

1. Run `HP Central AD Agent Setup.exe` that is located in HP Central AD agent installation folder, enter the Admin User account with built-in administrator privileges and the password:



2 Click **Install** button to start installation:



The status bar shows installation progress.

3 When installation is completed, click **OK** to exit:

4 After finishing with all necessary operations, you MUST reboot every domain controller manually to enable the Password Plug-In if you did not select Automatically Reboot Domain Controller before you start installation.

After the installation is finished, you can find the following items on the domain controller running HP Central AD agent:

- Three log files are added to *<Installer Dir>*`\HP_Central_AD_Agent\Log` folder:
  - `Reached.txt` – List machine names of all the reached domain controllers that have Password Plug-In installed successfully.
  - `Unreached.txt` – List machine names of all the unreached domain controllers that need to have Password Plug-In installed manually.
  - `LogInfo.txt` – List log messages.
- Data folder is added to the installation folder including following files:
  - `ADProperties.ini`, `ADPassfilt.dll`, `libeay32.dll` and `libssl32.dll` – These files are copied from `%SystemRoot%\System32` directory.
  - `DC_List.txt` – List names of all domain controllers it reached.
  - `DCFull_List.txt` – List full names of all domain controllers it reached.

On the target AD domain controller servers on which the Password Plug-In is installed successfully, you can find:

- Log folder is created as specified as `<PSLog_Path>` in `ADProperties.ini`.
- `ADProperties.ini`, `ADPassfilt.dll`, `libeay32.dll` and `libssl32.dll` are copied to `%SystemRoot%\System32` directory. And the following LDAP information is added into `ADProperties.ini`:

```
PSSync_Base_Suffix=DC=root, DC=sicf        '(Target DC's Domain Name)
PSSync_Server_Name=rootdc1.root.sicf       '(Target DC's Full DC Name)
```

- String "`ADPassfilt`" is appended to "`Notification Packages`" under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa` of the Registry.

For Central AD Agent,

- With Enterprise Administrator account, you can install the password plug-in on each domain controller.
- With domain administrator account, you can only install the password plug-in on the domain controller in the same domain.
- With built-in administrator account, you can only install the password plug-in on local machine.
- With other accounts, you can not install password plug-in on any domain controller.

5 Configuring the Connector with Select Identity

This chapter describes the procedure to configure the Active Directory Bidirectional LDAP connector with Select Identity and the connector specific parameters that you must provide while configuring the connector with Select Identity.

Configuration Procedure

After you deploy the connector RAR on application server, you must configure the connector with Select Identity. Perform the following steps to configure the Active Directory Bidirectional LDAP connector with Select Identity.

1. Add a New Connector
2. Add a New Resource
3. Map Attributes
4. Configure Workflow External Call on Select Identity
5. Configuring Exchange Related Attributes

Add a New Connector

Add a new connector in Select Identity by using the user interface. While adding the connector, do the following:

- In the Connector Name text box, specify a name for the connector.
- In the Pool Name text box, enter **`eis/ActiveDirConnector`**.
- Select **`No`** for the Mapper Available section.

Refer to the *HP Select Identity Connector Deployment Guide* for detailed information on adding a new connector in Select Identity.

Add a New Resource

Add a new resource in Select Identity that uses the newly added connector. Refer to the *HP Select Identity Connector Deployment Guide* for detailed instruction on adding a resource in Select Identity.

Refer to the following table while entering the parameters in the Basic Information and the Access Information pages:

Table 6 Resource Configuration Parameters

Field Name	Sample Values	Description	Comment
Resource Name	ELDAPADsample	Name given to the resource.	
Connector Name	ELDAPADsample	The newly deployed connector.	
Login Name	CN=Administrator, CN=Users,DC=sis, DC=com	Admin User Login Name.	If the Admin User cannot find the deleted users when performing Reconciliation, he will need to check the URL below for troubleshooting information: **http://support.microsoft.com/kb/892806/en-us**
Password		Password of the admin user.	If two-way authentication is enabled, then Login Name and Password will not be used. *When moving user across domain, make sure that the password complies with AD password complexity requirement.*
Mapping File	ActiveDir.xml	Name of the file that specifies the attribute mappings. This file should exist in the classpath of the application server. Click **View** to open the file in a browser. If this file cannot be viewed, Select Identity could not locate it.	
configFile	ActiveDirConfig	It contains configuration information and information of the entire forest. Specific information varies with customer environment.	

Table 6 Resource Configuration Parameters (cont'd)

Field Name	Sample Values	Description	Comment
objectClass	User	Entity type to provision. Each resource only supports one of the two entity types (contact or user).	You can only set the value to either user or contact.
Select Identity Locale	en_US	Locale-specific information. If Country=US and Language=English, current locale string is en_US.	
encryptionKey	6PqwwkfRTxaEJgW/cFuIUA==	Copy the key generated by password plug-in installer program.	
CRL Flag	false	Indicates if the resource performs CRL check. This flag works with CRL check flag in **Tools** → **System Security** → **Security Setup** → **Certificate Policy** page. If these two flags are both true, the connector will perform CRL check.	
Usage Flag	false	Indicates if the connector performs usage check. This flag works with usage check flag in **Tools** → **System Security** → **Security Setup** → **Certificate Policy** page. If these two flags are both true, the connector will perform Usage check.	
Delete Group Detection	false	Indicates if the connector supports deleted group reconciliation detection.	Not available in the current connector version.

Configuring Polling for Reverse Synchronization:

After entering the resource access information, User Reconciliation Policy page appears. On this page, do the following.

a Check the Polling Enable checkbox. Set the polling interval to the desired value.

b Under the Modify sections, set Reconciliation Workflow as Select Identity Recon User Enable Disable Workflow by using the drop-down box.

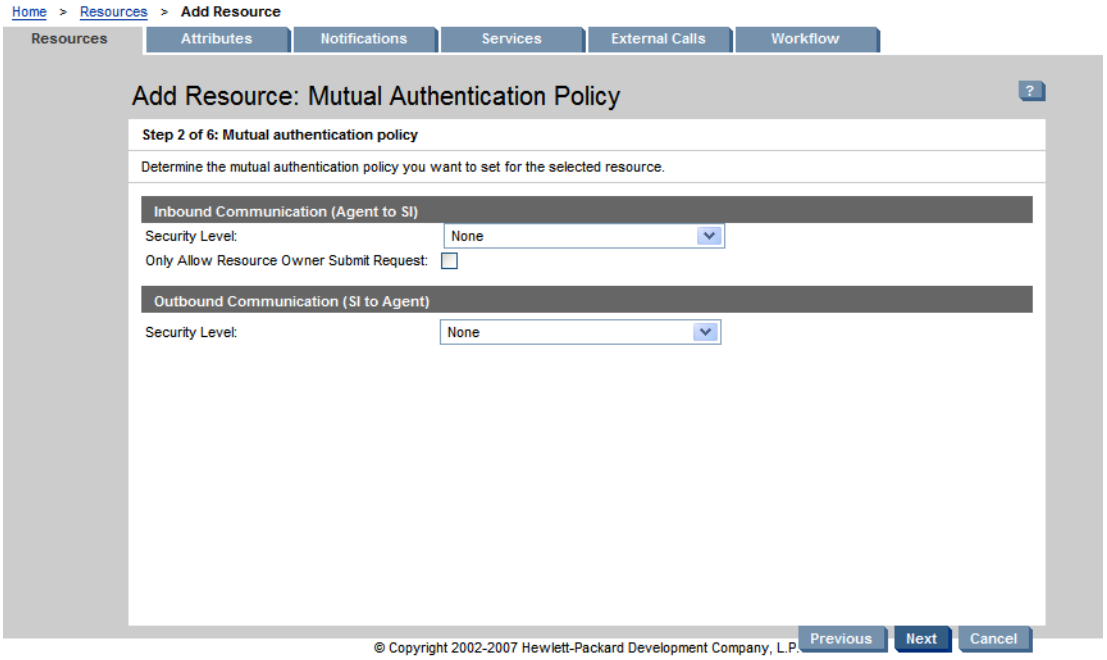
c Keep all other default settings in this page.

Configure for Mutual Authentication Support

In addition to common configuration (configure keystore and trust store properties into Security Level of Select Identity on page 26), some special configuration is needed in order to support Mutual Authentication for Active Directory.

Perform the following steps:

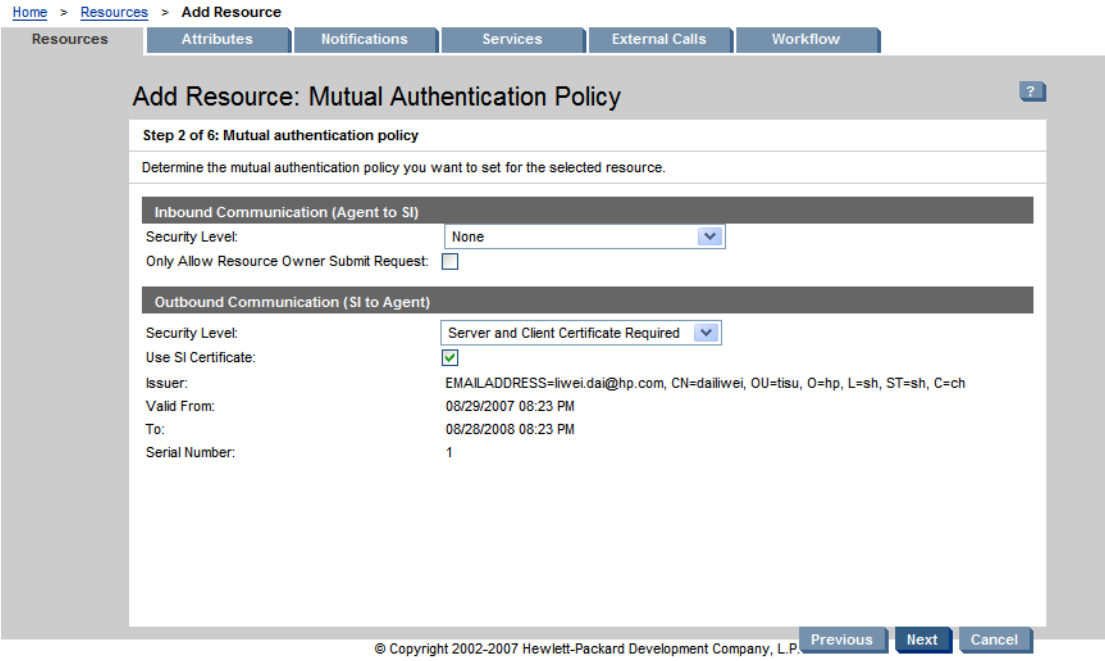
1 When adding a resource, on the **Add Resource: Mutual Authentication Policy** page, you can specify a mutual authentication policy by specifying the inbound and outbound security settings.



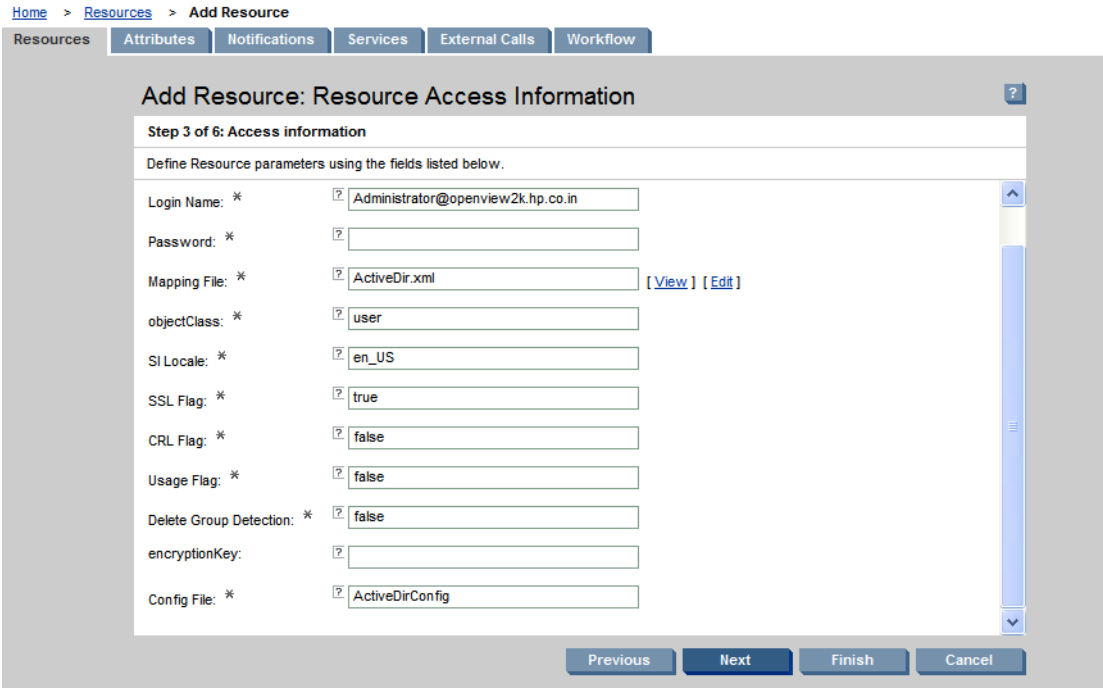
2 If you want to use one-way authentication between Select Identity and resource, select **Server Certificate Required** from the **Security Level** dropdown list in **Outbound Communication (SI to agent)** section.

If you want to use two-way authentication between Select Identity and resource, select **Server and Client Certificate Required** from the **Security Level** dropdown list in Outbound Communication (SI to agent) section, and make **Use SI Certificate** checked. Then, certificate information of Select Identity displays:

None value for Security Level is not applicable.



3 Click **Next**.



Attributes Login Name and Password are not used for mutual authentication if two-way authentication is selected in previous page.

Note that three new fields (CRL Flag, Usage Flag, and Delet Group Detection) are added:

— If CRL Flag is set to true, and Certificate Usage Validation is checked in **Tools** → **System Security** → **Security Setup** → **Certificate Policy** page, then CRL Validation is enabled.

— If Usage Flag is set to true, and CRL Validation is checked in **Tools** → **System Security** → **Security Setup** → **Certificate Policy** page, then Certificate Usage Validation is enabled.

Map Attributes

After successfully adding a resource for the Active Directory Bidirectional LDAP connector, make sure to map the resource attributes to Select Identity attributes. Refer to the *HP Select Identity Connector Deployment Guide* for information about mapping and creating attributes. While mapping attributes, refer to the following table for resource specific mapping information.

In order to support contact, now a new attribute entityType is available in user's memberAttributes definition, which is used to differentiate user and contact: If this attribute only belongs to user, you need to set `"entityType=`**`user`**`"`; If this attribute only belongs to contact, set `"entityType=`**`contact`**`"`; If this attribute belongs to user and contact, then set `"entityType=`**`user|contact`**`"`.

Table 7 Active Directory Bidirectional LDAP Mapping Information

Select Identity Resource Attribute	Connector Attribute	Attribute on Active Directory	Description
Street	streetAddress	streetAddress	*entityType= user | contact*
PhHome	homePhone	homePhone	*entityType= user | contact*
Email	Mail	mail	*entityType= user | contact*
PhMobile	mobile	mobile	*entityType= user | contact*
UserName	sAMAccountName	sAMAccountName	*entityType= user* *This attribute is mandatory for user creation.*
CN	cn	Cn	*entityType= user | contact* *This attribute is mandatory for user creation.*
Zip	postalCode	postalCode	*entityType= user | contact*
PhBus	telephoneNumber	telephoneNumber	*entityType= user | contact*
Password	unicodePwd	unicodePwd	*entityType= user* *This attribute is mandatory for user creation.*
Title	title	title	*entityType= user | contact*
DisplayName	displayName	displayName	*entityType= user | contact*
LastName	sn	Sn	*entityType= user | contact* *This attribute is mandatory for user creation.*

Table 7 Active Directory Bidirectional LDAP Mapping Information (cont'd)

Select Identity Resource Attribute	Connector Attribute	Attribute on Active Directory	Description
ObjectGUID	objectGUID	objectGUID	*entityType= user | contact* *This attribute is mandatory for user creation.* While associating Active Directory Bidirectional LDAP resource to a service, do not add this attribute to the service.
Groups	memberOf	memberOf	*entityType= user | contact*
FirstName	givenName	givenName	*entityType= user | contact*
UserPrincipalName	userPrincipalName	userPrincipalName	*entityType= user*
State	st	St	*entityType= user | contact*
Usersuffix	userSuffix	userSuffix	*entityType= user | contact* *This attribute is mandatory for user creation, and a valid value must be provided.* ***If UserSuffix needs to be configured as Select Identity service Fixed Attribute, make sure the value is all lower case.***

Table 7 Active Directory Bidirectional LDAP Mapping Information (cont'd)

Select Identity Resource Attribute	Connector Attribute	Attribute on Active Directory	Description
Domain	domain	domain	*entityType= user | contact* *This attribute is mandatory for user creation.* In a multi-domain environment, there may have more than one domain in the forest. Therefore, it is necessary to specify which domain a current operation will assign to. If one domain is specified, the operation will only assign the domain. Make sure to configure this attribute if you want the connector to work well as expected. ***If Domain needs to be configured as Select Identity service Fixed Attribute, make sure the value is all lower case.*** *If migrating the connector from v1.x to v2.x, the attribute name must be in all lower case, i.e., domain.*
City	l	L	*entityType= user | contact*
POBox	postOfficeBox	postOfficeBox	*entityType= user | contact*
userAccount Control	userAccount Control	userAccount Control	*entityType= user* While associating Active Directory Bidirectional LDAP resource to a service, do not add this attribute to the service.

The userSuffix specifies a place where the user is stored in the domain controller. If the userSuffix is empty, the connector will use the default userSuffix defined in the property files. For example, if you input the userSuffix as: ***ou=test,ou=selectidentity,ou=openview***, the user will be created in the OU in the Domain Controller.

If you modify the schema file (`ActiveDir.xml`), make sure that resource key is set to `objectGUID`.

Map the following attributes, if you want to provision users in Exchange mailbox.

Table 7A Exchange Mapping Information

Select Identity Resource Attribute	Connector Attribute	Attribute on Active Directory Bidirectional LDAP	Description
Email	Mail	mail	*entityType= user*
MailBoxStore	homeMDB	homeMDB	*entityType= user*
mailNickName	mailNickname	mailNickname	*entityType= user*
AlternateRecipient	altRecipient	altRecipient	*entityType= user*
HomeDirectory	homeDirectory	homeDirectory	*entityType= user*
AddressBook	showInAddressBook	showInAddressBook	*entityType= user*

Configure Workflow External Call on Select Identity

To achieve reverse synchronization, you must configure the workflow external call for user enable/ disable operation for Active Directory Bidirectional LDAP connector. When a user is enabled or disabled on resource (Active Directory), a specific Active Directory attribute value (`PSSync_ATTRIBUTE`) changes. The connector detects the change in the attribute value and registers the event as a user modification.

Refer to the *HP Select Identity Deployment Guide* for information about configuring user enable/disable workflow external call. While configuring, enter the parameters as given in Table 8 below.

Table 8 User Enable/Disable Parameters for Active Directory Bidirectional LDAP Connector

Serial Number	Parameter Name	Parameter Value
1.0	AttributeName	userAccountControl
2.0	EnableValue	512
3.0	DisableValue	514
4.0	UserName	Select Identity administrative user name. For example, sisa.
5.0	Password	Select Identity administrative password. For example, abc123.
6.0	Url	Select Identity web service url. For example: http://localhost:7001/lmz/webservice

While entering these parameters, check the Sensitive checkbox only in the case of Password.

Configuring Exchange Related Attributes

You can provision users in Exchange mailbox by using this connector. To be able to do that, you must map the exchange related attributes. These attributes are described below with example attribute values, which has to be entered during user provisioning.

- `Mail` — This is the Email Address for the user. For example, *user01@sitest.com*
- `homeMDB` — This is the ExchangeFolderDN and is a concatenation of several server values. For example, Example:

  CN=Mailbox Store (TLNT3),CN=First Storage Group,CN=InformationStore,CN=TLNT3,CN=Servers,CN=First Administrative Group,CN=Administrative Groups,CN=SITestOrg,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=sitest,DC=com

  This is a test DN. You must give an equivalent value.
- `mailNickname` – This nick name can be `User name` or `sAMAccountName`. For example:

  User01nick

  While adding user if you enter this value, email id of the user becomes - *User01nick@sitest.com*
- `altRecipient` — This is DN of any other User entry and used for forwarding mails from User01 to User02. For example, *CN=User02,CN=Users,DC=sitest,DC=com.*

  If you configure this attribute, then any mail that is sent to User01 will be forwarded to User02.
- `homeDirectory` — This is the virtual home folder. This is the location on which the Exchange User home directory will be stored. For example: *D:\temp*

  This folder is just shown as the User attribute and the folder is not created physically on the server.
- `showInAddressBook` — This is a concatenation of several server values. For example,

  CN=All Users,CN=All Address Lists,CN=Address Lists Container,CN=SITestOrg,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=sitest,DC=com | CN=Default Global Address List,CN=All Global Address Lists,CN=Address Lists Container,CN=SITestOrg,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=sitest,DC=com

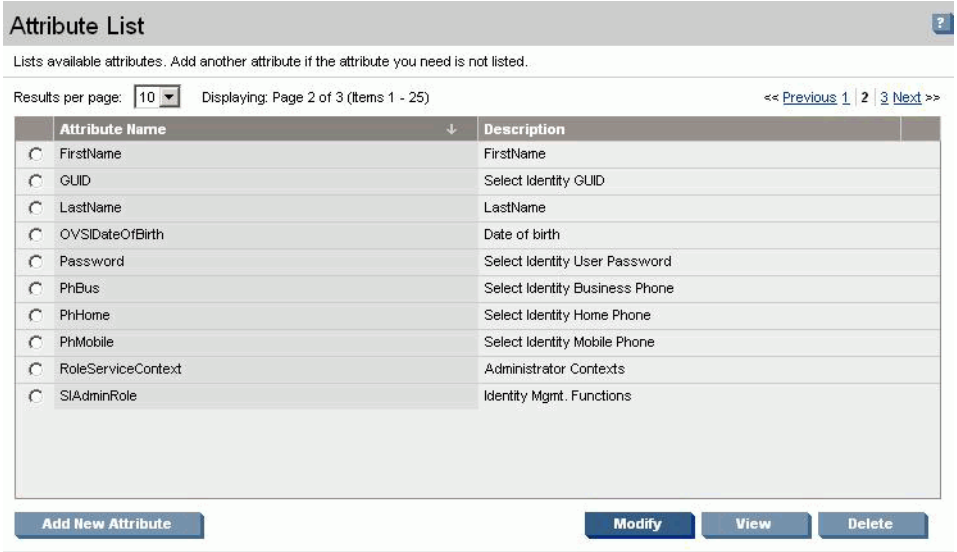
  This is a test value, you must give an equivalent value.

Configuring Password Expiry Operation

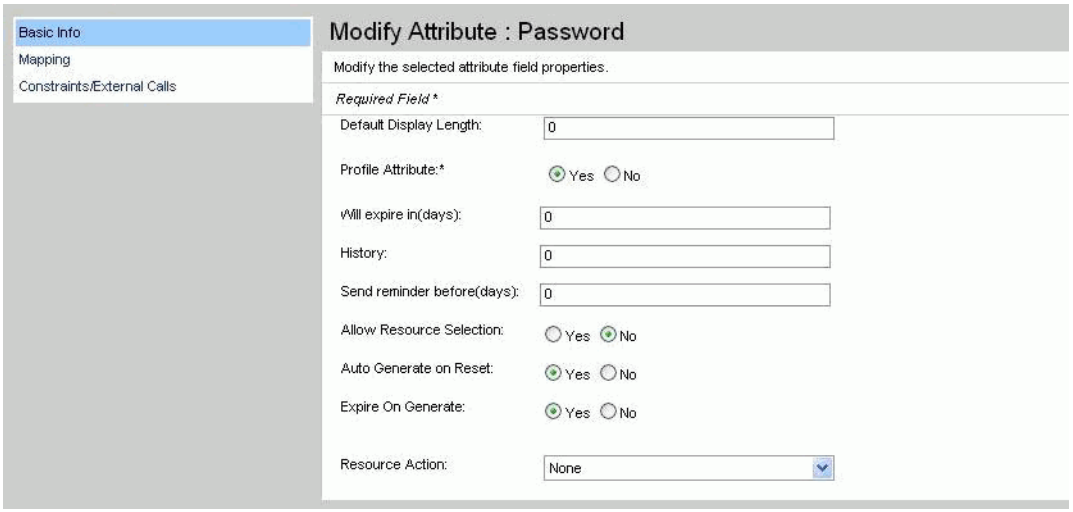
You can configure Select Identity to automatically expire the password (that has been automatically generated by Select Identity during user creation) of a newly created user.

Perform the following steps to configure password expiry operation on Select Identity 4.0-4.20:

1 In the Select Identity home page, click **Service Studio** → **Attributes**. The attributes list appears.



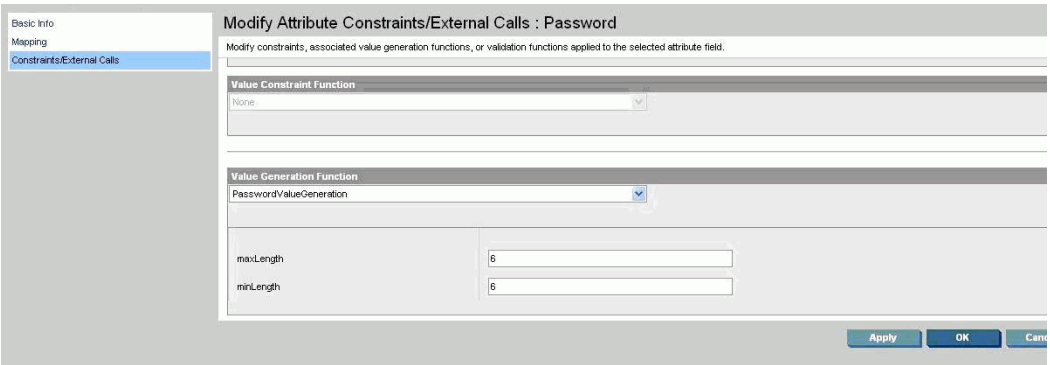
2 Select the `Password` attribute and click **Modify**. The Modify Attribute: Password page appears.



3 Select Yes in the Expire on Generate field.

4 Select Yes in the Auto Generate on Reset field.

5 Click the **Constraints/External Calls** link in the left pane. The Modify Attribute Constraints/External Calls : Password page appears.



6 From the Value Generation Function drop down box, select PasswordValueGeneration.

7 Click **Apply**.

➤ `Password` attribute should not be included in the Service form.

8 Open the schema file (`ActiveDir.xml`) by using a text editor and verify if the following XML string is present in the User section:

```
<attributeDefinitionReferenceattrFunction="provision|post|pre"attributeTy
pe="Read/write"
concero:isKey="false"concero:resfield="pwdLastSet"concero:tafield="{0}"
defaultValue="0"encrypt="false" encrypted="false"
encryptionAlgorithm=""expirePassword="true" expireValue="0"
isPassword="false"linktoentity=""
multivalued="false"mustOnResource="false"name="objectclassuserattributepw
dLastSet"objectclass="user" objectclasstype="structural"ordering=""
remexpireValue="-1" renamekey="false"required="false"
resourcekey="false"supportedOperations="UNLINK,LINK,GETATTRIBUTES,GETPARE
NT,GETCHILDREN,GETALL,RESETPASSWORD,CHANGEPASSWORD,EXPIREPASSWORD,DISABLE
,ENABLE,CREATE,DELETE,UPDATE"transform="NO" type="java.lang.String"/>
```

6 Uninstalling the Connector

If you want to uninstall the connector, perform the following steps:

- Remove all resource dependencies in Select Identity.
- Delete the connector from Select Identity.
- Delete the connector from application server.
- Run the Password Plug-In Wizard on the domain controller to uninstall password plug-in.
- If HP Central AD Agent is installed in a multi-domain environment, you can run HP Central AD Agent from the server it is installed to automatically remove the password plug-ins on all other domain controllers.

See *HP Select Identity Deployment Guide* for more information about deleting the connector from application server and Select Identity.

A Troubleshooting

- While creating the user if the password is not set and an exception with `5003` code is thrown.

  Solution:

  Verify whether the password sent to the user meets the password policy.

  For example, the default password policy should accept a password with 8 or 9 characters with at least one uppercase and a numeric value (Password1).

- While creating and trying to save a resource, you get error `The following resource failed to save: Reason: Unable to test connector.`

  Solution:

  Verify if the following config file is in the application server classpath while deploying the connector.

  — `com\hp\ovsi\connector\bidirldap\activedir\ActiveDirConfig.properties`

- Bypassing of Link/Unlink operation does not work.

  Solution:

  In the `ActiveDirConfig.properties` file, set the `dualLink-support` parameter to 2 and ensure `byPass` is configured for both the `User` and `Group/Computer` entities in the connector schema file.

- Communication exception occurs with WebSphere when user operations are tried after a brief pause and the following error message appears in the log file:

  `javax.naming.CommunicationException`

  Cause:

  Connection timeouts of JCA connections in the applications server do not match with the connection timeout of the connector with the resource.

  Solution:

  In Active Directory, the resource time out (`MaxConnIdleTime`) should be greater than sum of Unused timeout and Reap time in WebSphere connection pool parameters. Also, the Minimum connections should be set to 0. Perform the following steps on WebSphere console to change the connection pool settings:

  a Log on to WebSphere console.

  b In the left pane, click **Resources** → **Resource Adapters**.

  c In the right pane, click on the connector name under the Preferences section.

  d In the right pane, click **J2C connection factories**.

  e Click on the connector name under the Preferences section.

  f In the right pane, click **Connection pool properties**.

g Under the General Properties section, make the following changes:

  – Set the Minimum connections to 0.

  – Set the Aged timeout to a value greater than 0.

  – Set the Reap time and Unused timeout in such a way that the sum of the Reap time and the Unused timeout is lesser than the value of `MaxConnIdleTime` on Active Directory server.

- Reconciliation fails occasionally.

  Solution:

  Make sure all resource attributes are mapped in SI.

- If password plug-in is uninstalled and then reinstalled, it will affect the existing users.

  Solution:

  When you reinstall the password plug-in, manually modify the key in `ADProperties.ini` file with the old key that is restored in the encryptionKey field of the Resource property in Select Identity.

  Or

  Do NOT select Generate a New Key when you reinstall the password plug-in.

- User link to group fails.

  In AD, there are three kinds of groups: Domain Local, Global and Universal. User can not be linked to the Global Group of a different domain.

- Reconciliation for deleted users fails.

  Solution:

  Make sure that a newly created user is pulled into the Select Identity server before you delete the user, otherwise the connector ignores the delete reconciliation.

  Or

  If you want to allow non-administrators to view deleted objects in Active Directory, make sure to modify the permissions on the deleted objects container, so that non-administrators can view this container by running DSACLS.exe which is included with the Active Directory Application Mode (ADAM) Administration Tools.

  After installation of ADAM Administration Tools, you can modify the permissions on the deleted objects container:

  a Log on with a user account that is a member of the Domain Admins group.

  b Click **Start** → **All Programs** → **ADAM** → **ADAM Tools Command Prompt**.

  The ADAM Tools Command Prompt window appears.

  c In the command prompt, type a command that is similar to the following example:

dsacls "CN=Deleted Objects,DC=root,DC=sicf" /g ROOT\ADAM_Test:LCRP

```
ADAM Tools Command Prompt
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\ADAM>dsacls "CN=Deleted Objects,DC=root,DC=sicf" /g ROOT\ADAM_Test:LC
RP_
```

When you type the command, make sure to use the name of the deleted objects container for your domain.

Each domain in the forest has its own container for deleted objects.

You can also copy ADAM installation folder to your target DC without installing ADAM Administration Tools. In the installation folder, you can find `DSACLS.exe` and type a command that is similar to the following example in command prompt:

dsacls "CN=Deleted Objects,DC=root,DC=sicf" /g ROOT\ADAM_Test:LCRP

Press Enter. The output window appears:

```
ADAM Tools Command Prompt
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              LIST CONTENTS
                              READ PROPERTY
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              READ PROPERTY
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              DELETE CHILD
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              DELETE CHILD
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              LIST CONTENTS
                              READ PROPERTY
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              LIST CONTENTS
                              READ PROPERTY
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              LIST CONTENTS
                              READ PROPERTY
Allow ROOT\ADAM_Test          SPECIAL ACCESS
                              LIST CONTENTS
                              READ PROPERTY

The command completed successfully

C:\WINDOWS\ADAM>_
```

User ROOT\ADAM_Test has been granted List Contents and Read Property permissions on the deleted objects container in the ROOT domain. These permissions allow the user to view the contents of the deleted objects container, but the user is not allowed to make any changes to objects in the container. These permissions are equivalent to the default permissions that are granted to the Administrators group.

- Attribute `domain` cannot be found in attribute list on Select Identity after Active Directory Bidirectional LDAP connector v2.0 or a later version is deployed.

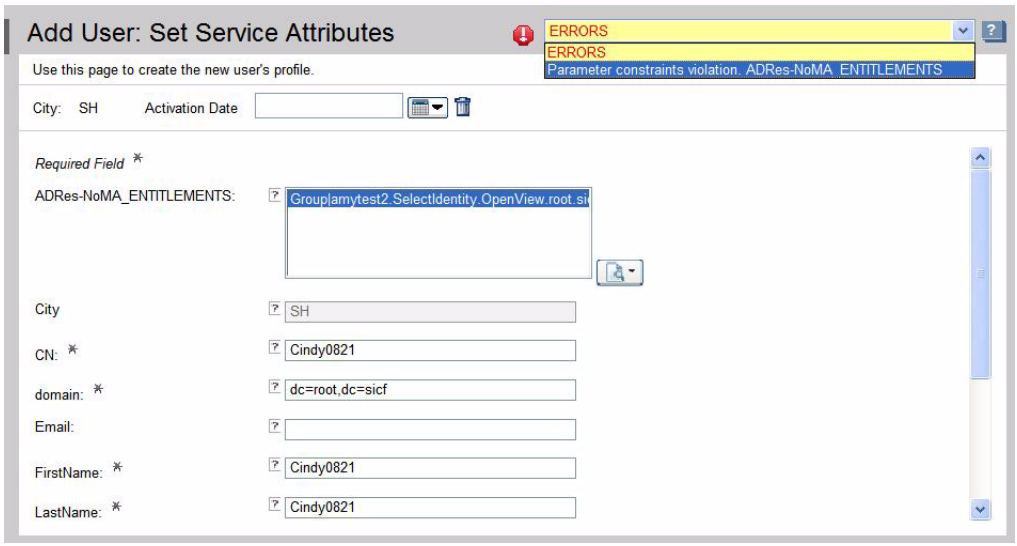
  Solution:

  For detailed information, see Verifying Attribute Addition/Deletion on Select Identity on page 93.

- After Select Identity version upgrade, the request of creating user with entitlement cannot be submitted successfully. You may see an error message similar to the following:

  ```
  ERRORS

  Parameter constraints violation. <Resource_ENTITLEMENT>
  ```

  Cause:

  The database script (`mssql_cbc_ddl.sql` for MS SQL database or `Oracle_cbc_ddl.sql` for Oracle) available in `cbc_config.zip` was not executed after Select Identity version upgrade.

  Solution:

  Re-execute the database script.

- CRL check fails.

  Solution:

  The Sun JDK version 1.5.0_06-b05 is too low, update to Sun JDK version 1.5.0_09-b03 or higher versions of 1.5.

- If userSuffix attribute is missing in Select Identity service, then reconciliation for group membership change in Webshpere with Windows 2000 AD Server will fail.

  Solution:

  Make sure that userSuffix attribute exists in Select Identity service, and a valid value is provided for it.

- Moving user across domain fails.

  Solutions:

  Perform the following steps:

  a Check if the Windows Native Function (WNF) framework is properly installed;

  b Make sure that the target OU exists;

  c Check if the `transientUserSuffix` attribute exists in AD server;

d Check the connector properties file (normally the `ActiveDirConfig.properties` file) to see if the request/response attributes match with the request/response attributes defined in `PasswordAgent-config.xml` on Agent side.

e If it is running in Mutual Authentication mode, check that the password for creating resource is valid for AD server.

- Creating resource fails in Mutual Authentication mode.

  Solutions:

  Perform the following steps:

  a Check if AD server is active and can be located and connected;

  b Make sure that Mutual Authentication has been properly configured in the Select Identity. Refer to

  c Check if the global catalog port is set to **`3269`** in the connector properties file.

- No reconciliation request to Select Identity.

  Solutions:

  Check if the OVSI_BIDIRLDAP_LCLN is properly configured, as shown in the example below:

	DNS_Name	HighestCommittedUSN	ResourceName
1	SICF-AD1.root.sicf	465493	ADResourceRootNoMA

- Verify that the Password Plug-In has been installed successfully.

  Solutions:

  Perform the following steps to verify:

  a Make sure that the following files exist in `System32` directory:

  `ADPassfilt.dll`
  `ADProperties.ini`
  `libeay32.dll`
  `libssl32.dll` (32bit AD server) / `ssleay32.dll` (64bit AD server)

  b Check in the Registry that string `ADPassfilt` exists in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages`.

- Verify that moving user across domain has been installed successfully.

  Solutions:

  Perform the following steps to verify:

  a Make sure that the following files exist in `System32` directory:

  `PasswordAgent-config.xml`
  `Interop.ActiveDs.dll`
  `log4net.dll`
  `HP.AD.Logging.config`
  `HP.AD.Common.Logging.dll`
  `HP.AD.WNF.ActionInterface.dll`
  `HP.AD.WNF.Delegate.dll`
  `HP.AD.WNF.MoveUser.dll`
  `HP.AD.WNF.Utilities.dll`

b Check in the Registry that `HP.AD.WNF.CommandDelegate` exists in `HKEY_CLASSES_ROOT`.

- Check the Password Plug-In and Windows Native Function framework version.

  Solutions:

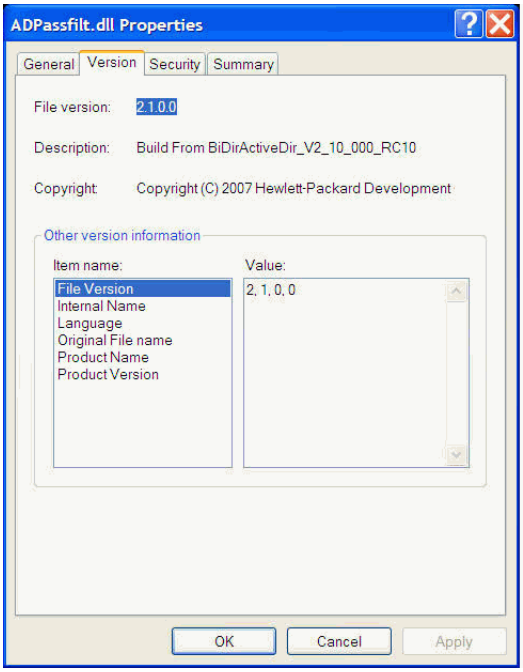
  — Password Plug-In:

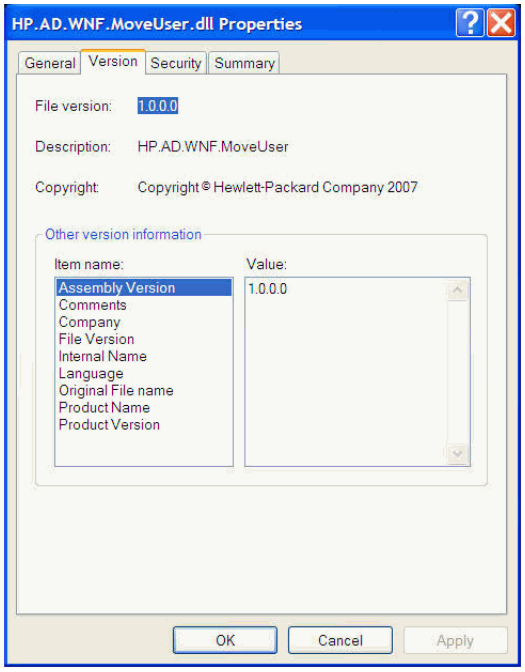
  Locate `ADPassfilt.dll` file in `System32` directory, right click on it and select **Properties** from the popup menu. The ADPassfilt.dll Properties windows displays. You can find the Password Plug-In version on the Version tab, as shown below:

  — Windows Native Function framework version:

  Moving user across domain function works through Windows Native Function (WNF) framework.

  To check WNF version, locate `HP.AD.WNF.MoveUser.dll` file in `System32` directory, right click on it and select **Properties** from the popup menu. The ADPassfilt.dll Properties windows displays. You can find the WNF version on the Version tab, as shown below:



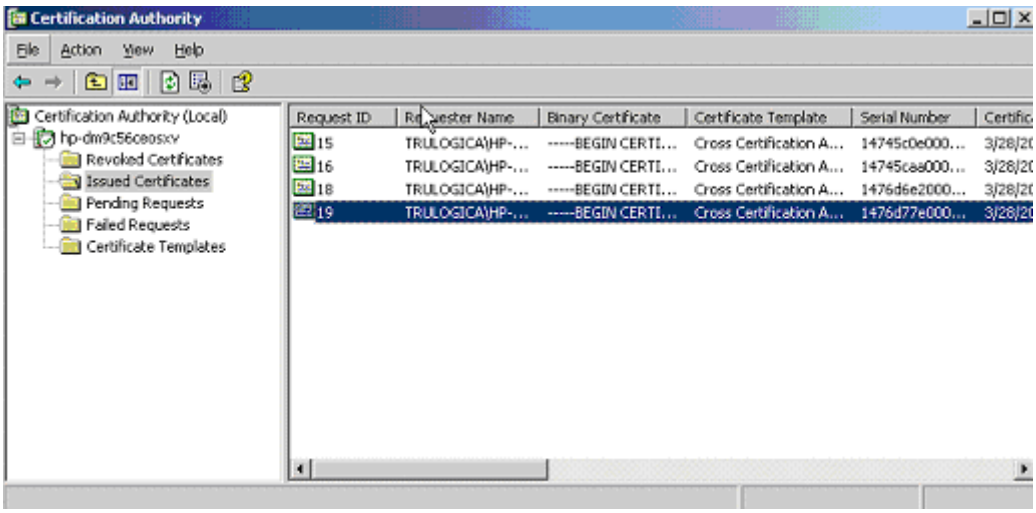


B Installing Certificate

Generating A Root CA Certificate on Active Directory

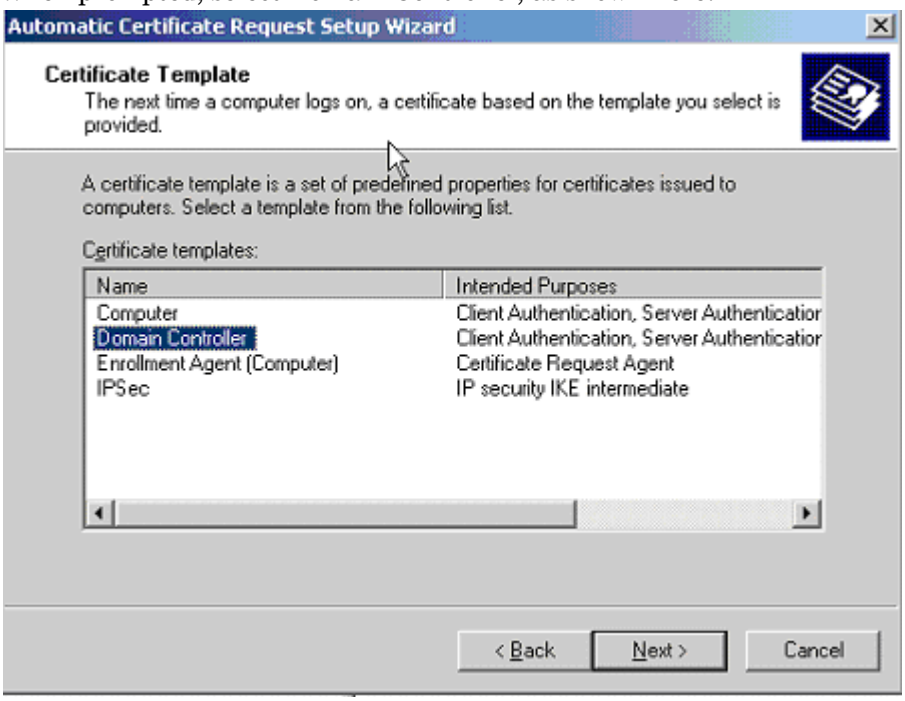
Perform the following steps to generate a Root CA Certificate on Active Directory:

1. Install the Certificate Services Component from the Windows CD.
2. Configure HTTPS on the system.
3. Create a Certificate Authority (from Administrative Tools ¨ Certification Authority), which also creates a root certificate. The following shows the certificate after it is created on Windows 2003:

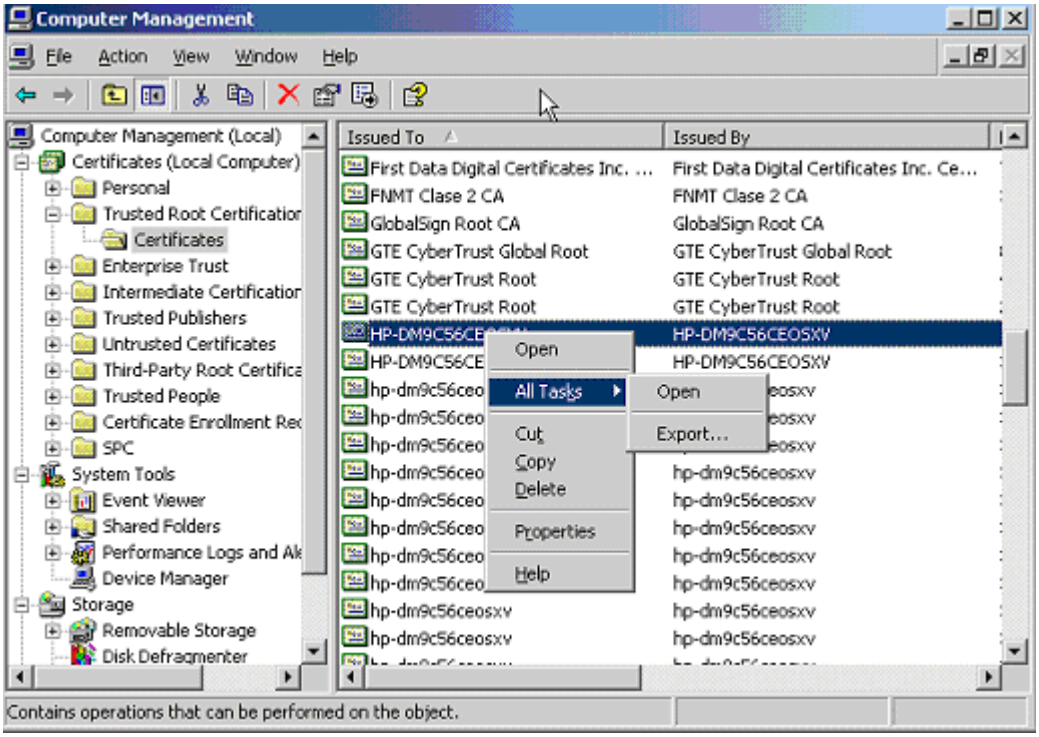


4. Create an Automatic Certificate Request (from **Administrative Tools** → **Domain Controller Security Policy** → **Public Key Policies**).

When prompted, select Domain Controller, as shown here:



5 After the new entries are displayed in Administrative Tools → Certification Authority → Issued Certificates, open the certificate (by using the snap-in from mmc), which is located under Trusted Root Certification Authorities → Certificates and has the same name as the CA.

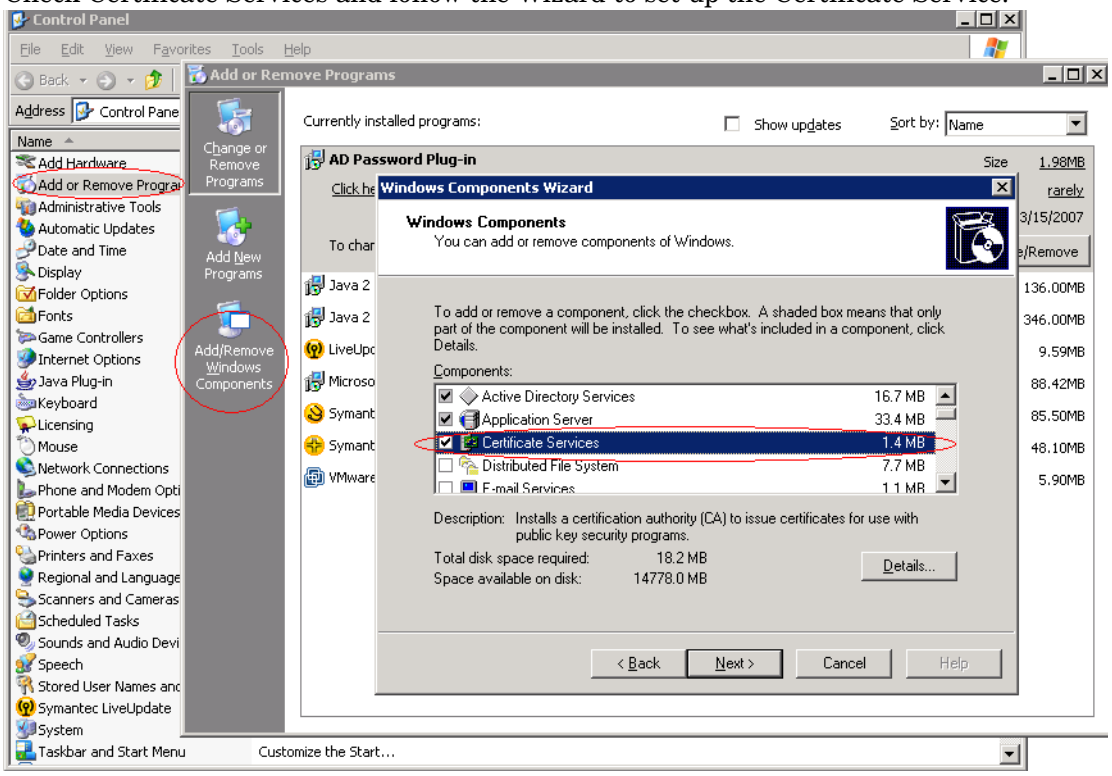


Export the certificate and specify a file name with the extension `.cer`.

Setting Up Certificate Service

Follow steps below to set up Certificate Services:

1. From the Start menu, click **Control Panel→Add or Remove Programs**. The Add or Remove Programs window opens.
2. Click **Add/Remove Windows Components** from left panel to start Windows Components Wizard.
3. Check Certificate Services and follow the Wizard to set up the Certificate Service.

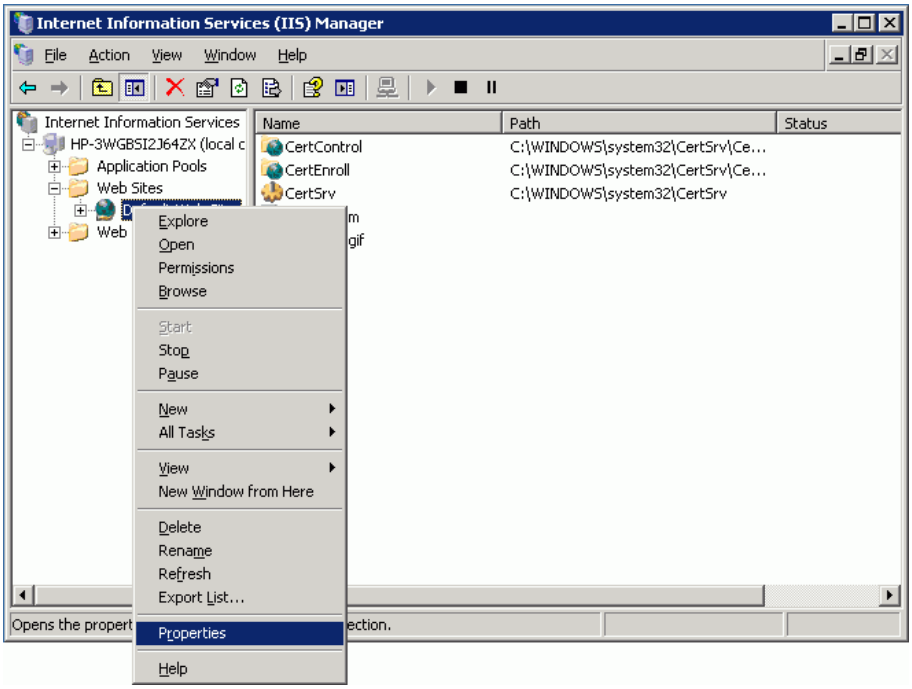


Generating Information for Applying for A New Certificate

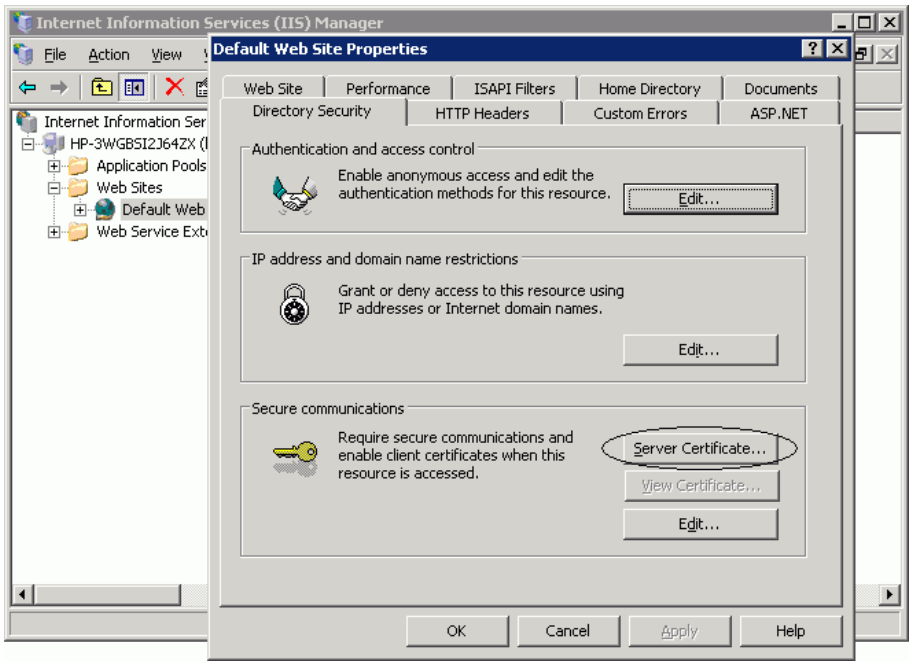
Follow steps below to generate information for applying for a new certificate:

1. On the Active Directory server, from the Start menu, click **Administrative Tools→Internet Information Services (IIS) Manager**. The Internet Information Services (IIS) Manager window opens.

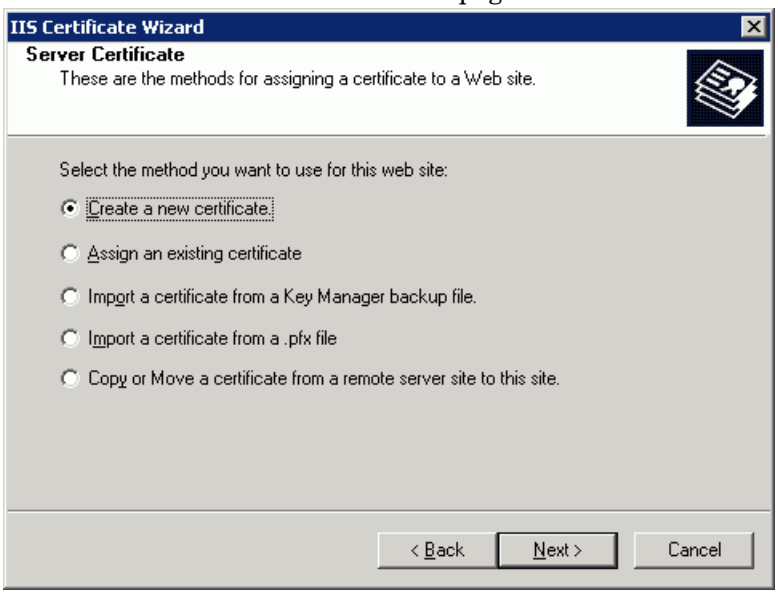
In the left panel, expand local computer node→**Web Sites**. Right click Default Web Site and select Properties from the context menu to open Default Web Site Properties window.



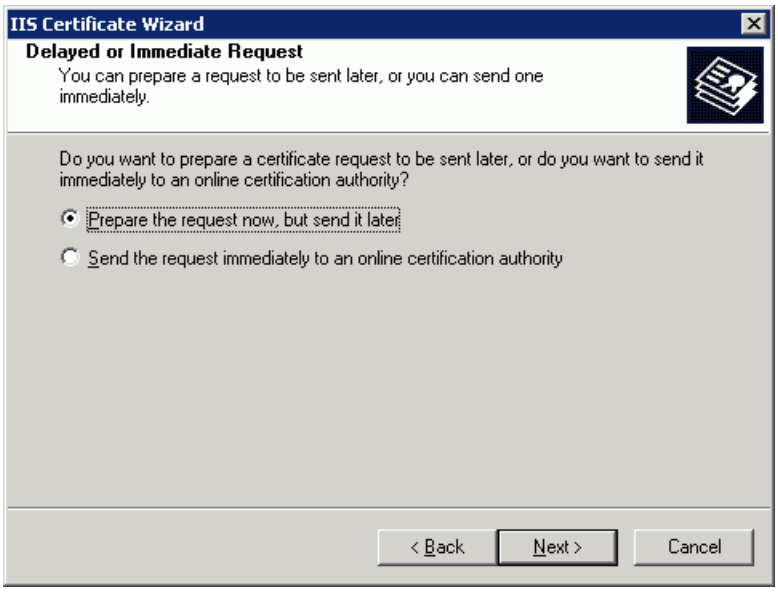
From Directory Security tab of Default Web Site Properties window, click **Server Certificate** to start Web Server Certificate Wizard.



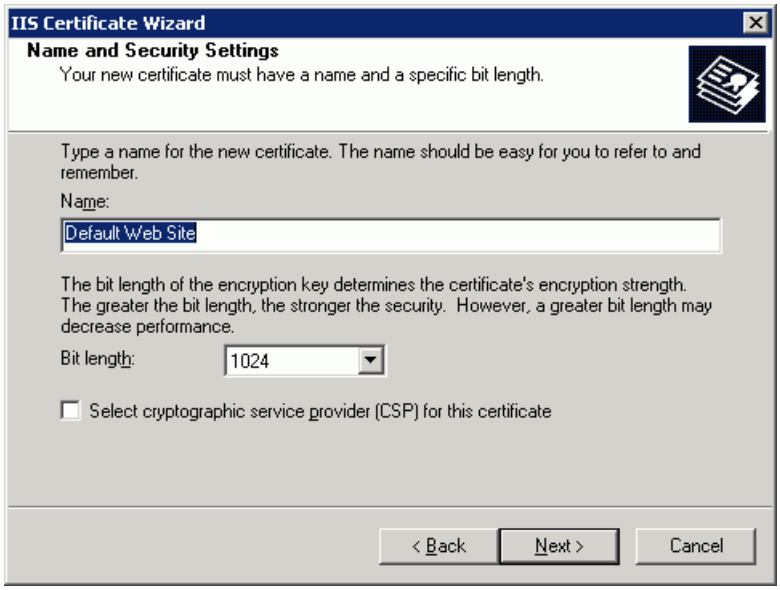
2 Click **Next** to enter Server Certificate page and select **Create a new certificate**.



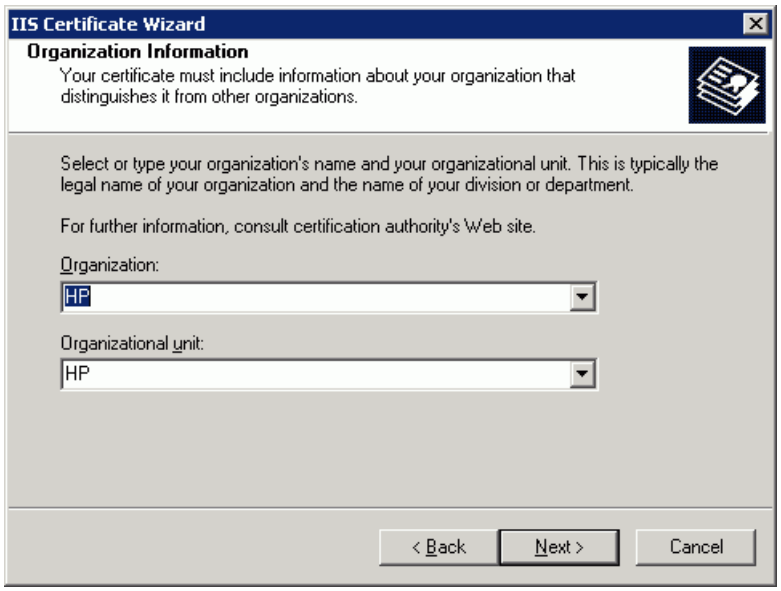
Click **Next** to enter Delayed or Immediate Request page, then select **Prepare the request now, but send it later.**



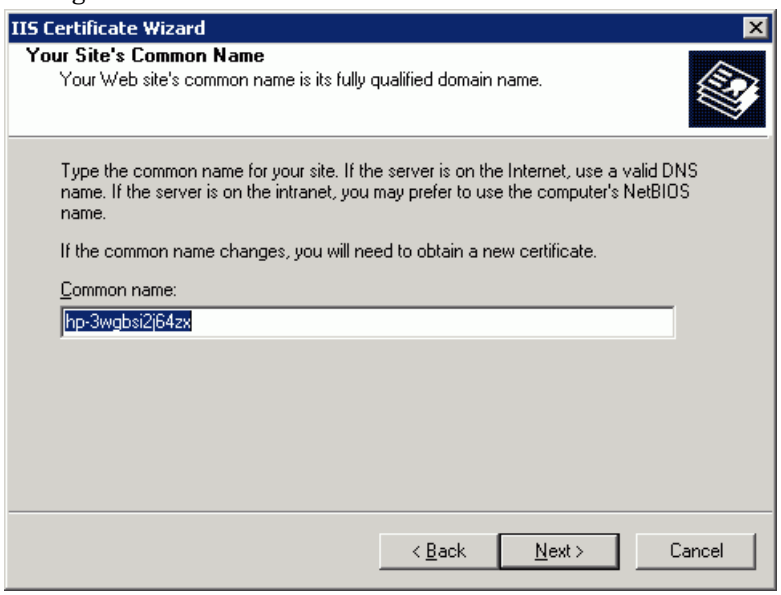
Click **Next** to enter Name and Security Settings page. Provide a name or keep the default setting as you like.



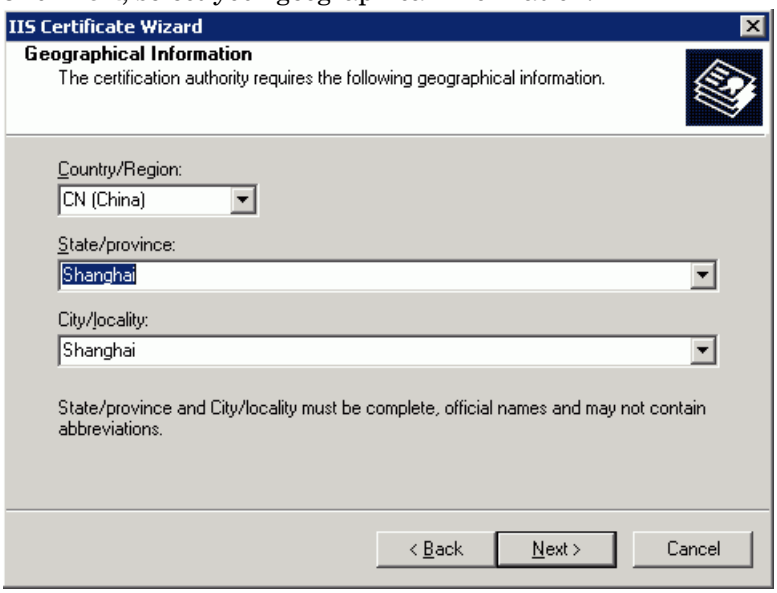
Click **Next** to enter next page. Provide necessary organization information as prompted.



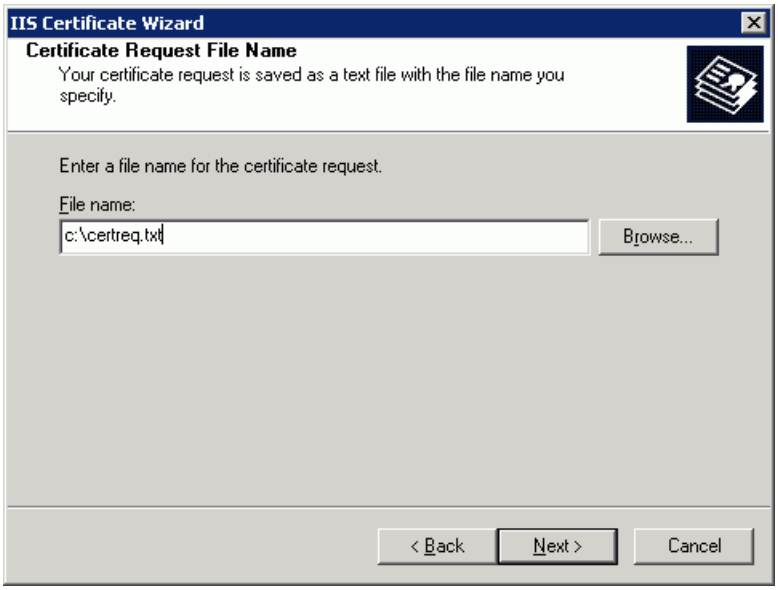
Click **Next** to enter next page. Provide a common name if you want, or keep the default setting.



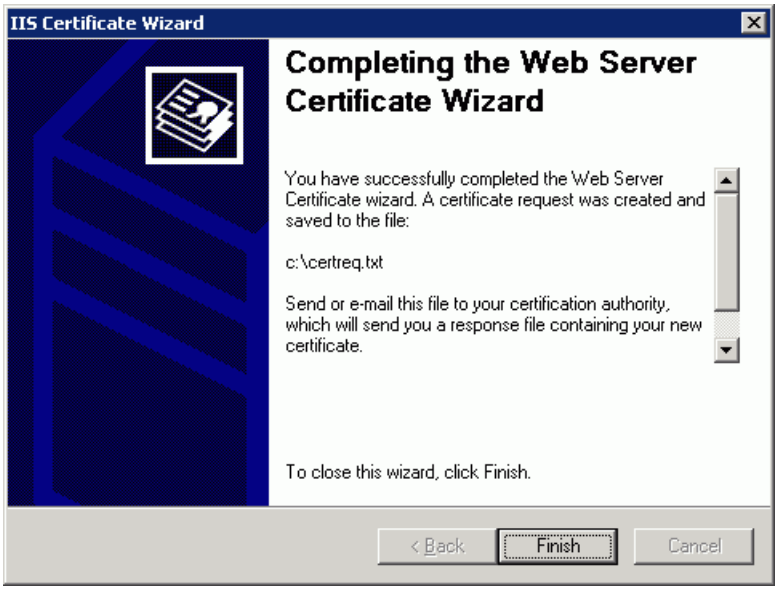
Click **Next**, select your geographical information.



Click **Next**. Provide a certificate name.



Click **Next**, check request file summary. Then click `Next` again.



Click **Finish**, the request information is saved in the text file: `c:\certreq.txt`.

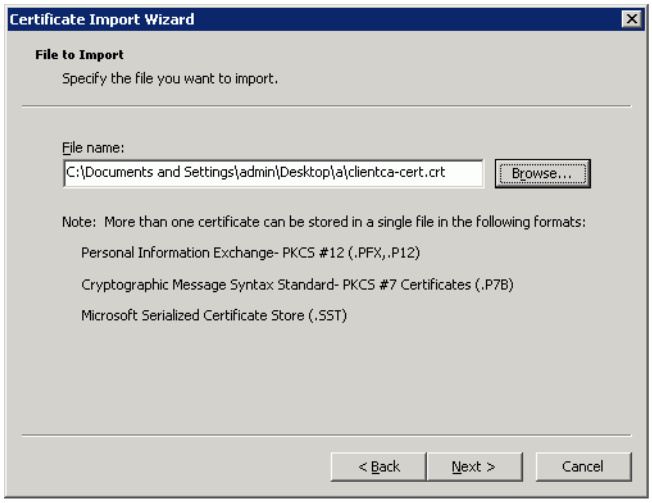
C Importing a Certificate into Active Directory Server

Manual configuration on Active Directory Server is required for SSL connection between the Select Identity and Active Directory server.

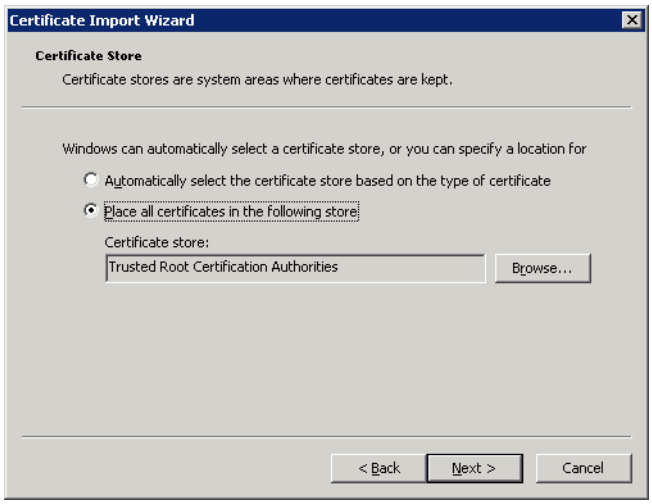
Importing a Certificate into Active Directory Computer's Trusted Root CA Certificate Store

Perform the following steps to import a certificate into AD computer's Trusted Root CA Certificate Store:

1. Enter `mmc` in Run box and click **OK** to launch MMC snap-ins.
2. Select **File** → **Add/Remove Snap-in**. The Add/Remove Snap-in window displays.
3. Click **Add**, the Add Standalone Snap-in window displays.
4. Select **Certificates**, then click **Add**. The Certificates snap-in window pops up.
5. Choose **Computer account**, then click **Next**. The Select Computer window displays.
6. With **Local computer** selected, click **Finish**.
7. Click **Close** in the Add Standalone Snap-in window.
8. Click **OK** in the Add/Remove Snap-in window.
9. In the MMC console, expand **Certificates (Local Computer)** → **Trusted Root Certification Authorities** → **Certificates**. Right click **Certificates**, and select **All Tasks** → **Import...**, the Certificate Import Wizard displays.
10. Click **Next**, the File to Import page displays. Locate the certificate:



11 Click **Next**. The Certificate Store page displays.



12 Click **Next**. Then click **Finish**. The import is successful.

Importing a Certificate into Active Directory Computer's Personal Certificate Store

Perform the following steps to import a certificate into AD computer's Personal Certificate Store:

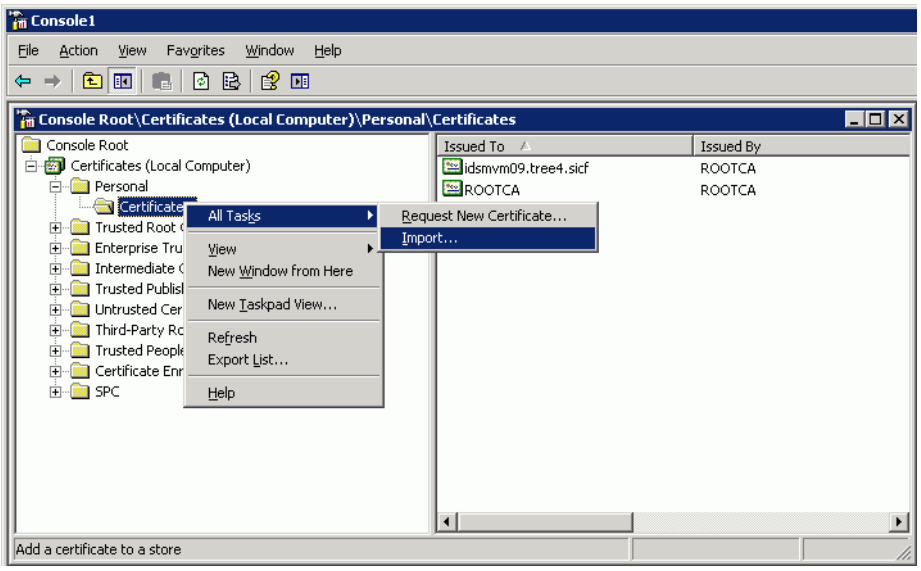
1 Get the ceritifate, for example, *thirdParty*.crt.

2 Use command to convert *thirdParty*.crt into *thirdParty*.pfx:

openssl pkcs12 -export -inkey server.key -in *thirdParty*.crt -out *thirdParty*.pfx

3 Import *thirdParty*.pfx into AD computer's Personal Certificate Store of resource.

In the MMC console, expand **Certificates (Local Computer)** → **Personal** → **Certificates.** Right click **Certificates** and select **All Tasks** → **Import**.

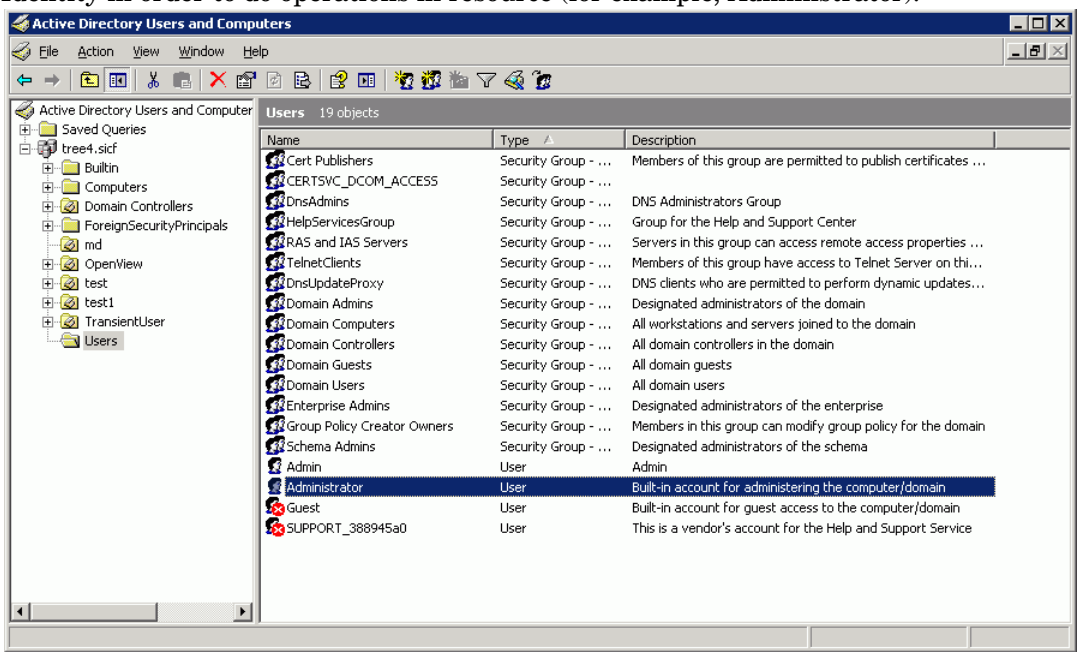


Repeat step 10 to step 12.

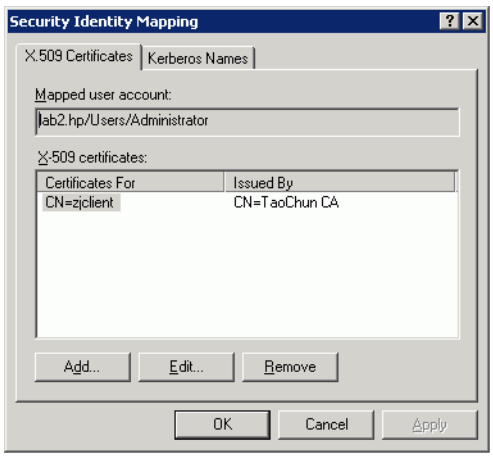
Mapping a User to Select Identity Certificate in AD

Make sure to select an option in AD, then perform the following steps to map a user to Select Identity certificate

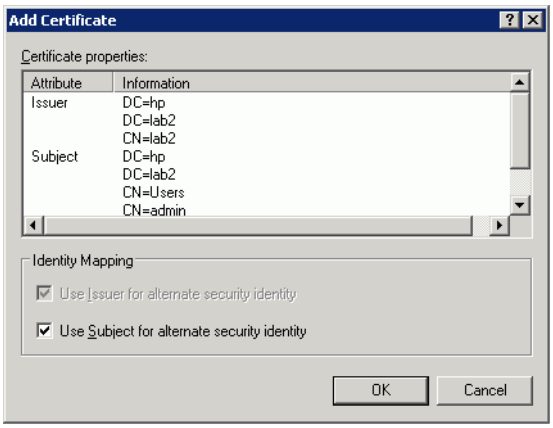
1. Open **Active Directory Users and Computers**. The Active Directory Users and Computers window displays.
2. Click **View** → **Advanced Features**.
3. Click **Users** node in the navigation pane, and select a user with access rights to Select Identity in order to do operations in resource (for example, Administrator).



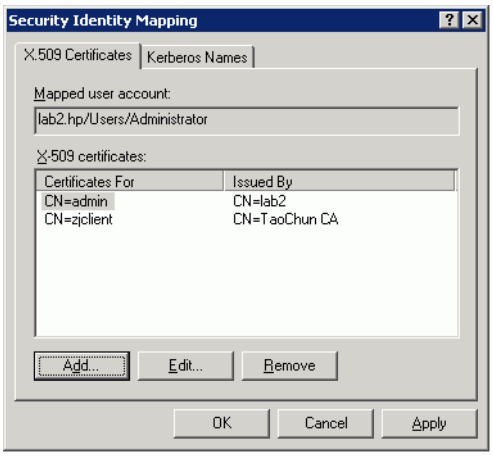
4 Right click the user and select **Name Mappings**. The Security Identity Mapping window displays.



5 Click **Add** and locate Select Identity certificate file:



6 Click **OK**.



7 Click **OK**. The user is mapped to Select Identity certificate.

D Customizing Schema File

In addition to properties files, such as `ActiveDirConfig.properties` file, there is also a schema file (`ActiveDir.xml`) present in `ActiveDirSchema.jar` file, which defines the relationship of attributes mapping between Select Identity and the connector. You may customize this schema file to meet your needs.

Adding New Attribute Mapping

To add a new attribute in the schema file, you need to add two tags:

1. First, add a new tag `<attributeDefinitionReference>` in `<Schema>\<objectClassDefintion description="" name="User">\<memberAttributes>` of the schema file.

```
+ <objectClassDefinition description="" name="Group">
+ <objectClassDefinition description="" name="Computer">
- <objectClassDefinition description="" name="User">
  + <properties>
  - <memberAttributes>
      <attributeDefinitionReference attrFunction="provision|post|pre" attributeType="Read/write" concero:isKey="false"
        concero:resfield="userAccountControl" concero:tafield="userAccountControl" defaultValue="" encrypt="false"
        encrypted="false" encryptionAlgorithm="" expirePassword="false" expireValue="" isPassword="false" linktoentity=""
        multivalued="false" mustOnResource="false" name="objectclassuserattributeuserAccountControl" objectclass="user"
        objectclasstype="structural" ordering="" remexpireValue="" renamekey="false" required="false" resourcekey="false"
        entityType="user"
        supportedOperations="UNLINK,LINK,GETATTRIBUTES,GETPARENT,GETCHILDREN,GETALL,RESETPASSWORD,CHANGEPASSWORD,EX
        transform="NO" type="java.lang.String" />
    - <!--
                <attributeDefinitionReference
                    attrFunction="provision|post|pre"
```

2. Add tag `<attributeDefinition>` in tag `<Schema>` of the schema file.

```
+ <objectClassDefinition description="" name="Group">
+ <objectClassDefinition description="" name="Computer">
+ <objectClassDefinition description="" name="User">
- <attributeDefinition description="Group_objectclassgroupattributemember" name="Group_objectclassgroupattributemember"
    type="java.lang.String">
  - <properties>
    - <attr name="minLength">
        <value>0</value>
      </attr>
    - <attr name="maxLength">
        <value>255</value>
      </attr>
    - <attr name="defaultValue">
        <value />
      </attr>
    - <attr name="pattern">
      - <value>
          <![CDATA[ [a-zA-Z0-9@]+  ]]>
        </value>
      </attr>
    </properties>
  </attributeDefinition>
- <attributeDefinition description="Group_objectclassldapv3ConnectorattributegroupSuffix"
    name="Group_objectclassldapv3ConnectorattributegroupSuffix" type="java.lang.String">
  - <properties>
```

3. Repeat the two steps above to add more attributes.

Below is an example of `<attributeDefinitionReference>` tag which describes parameters of the attribute:

```
<attributeDefinitionReference attrFunction="provision|post|pre"
attributeType="Read/write" concero:isKey="false"
concero:resfield="userAccountControl" concero:tafield="userAccountControl"
defaultValue="" encrypt="false" encrypted="false" encryptionAlgorithm=""
expirePassword="false" expireValue="" isPassword="false" linktoentity=""
multivalued="false" mustOnResource="false"
name="objectclassuserattributeuserAccountControl" objectclass="user"
objectclasstype="structural" ordering="" remexpireValue="" renamekey="false"
required="false" resourcekey="false" entityType="user"
supportedOperations="UNLINK,LINK,GETATTRIBUTES,GETPARENT,GETCHILDREN,GETALL,
RESETPASSWORD,CHANGEPASSWORD,EXPIREPASSWORD,DISABLE,ENABLE,CREATE,DELETE,UPD
ATE" transform="NO" type="java.lang.String" />
```

You can modify values of the parameters of the attribute according to their descriptions:

- `attrFunction="provision|post|pre"`

  This parameter specifies provisioning types of the attribute. String `"provision|post|pre"` is used as the default value which is three workflows on Select Identity: provision, post provision, and pre provision. You can choose workflows the attribute supports by modifying the attribute value.

- `attributeType="Read/Write"`

  This parameter specifies whether the attribute is allowed to have Read/Write permissions on resource. String `"Read/Write"` is used as the default value, which means the attribute can be read and write on resource. Make sure to set the value of this parameter according to permission of the attribute on resource.

- `concero:isKey="false"`

  This parameter specifies whether the attribute is a key to uniquely identify an object on Select Identity.

  Among all attributes in the schema file, only one attribute can be specified to have `true` value.

- `concero:resfield="userAccountControl"`

  This parameter specifies the attribute name, which corresponds to the attribute on the resource.

- `concero:tafield="userAccountControl"`

  This parameter specifies the attribute name, which corresponds to the attribute on Select Identity.

- `defaultValue=""`

  This parameter specifies the default value.

- `encrypt="false"`

  This parameter specifies whether encryption is required for the attribute. If parameter `encrypt` is true, the value of the attribute need be encrypted by the Connector.

- `encrypted="false"`

  This parameter specifies whether the value coming from Select Identity is in encrypted state. If parameter `encrypted` is true, the value of attribute need not be encrypted by the Connector again.

- `encryptionAlgorithm=""`

If parameter `encrypt` is set to `true`, this parameter specifies the algorithm used for encryption.

- `expirePassword="false"`

  This parameter is used with `expireValue` to set password expired.

- `expireValue="-1"`

  This parameter specifies the default value to set password expired. If `expirePassword` is `true` and `expireValue` is `-1`, password is set expired.

- `isPassword="false"`

  This parameter specifies whether this attribute is a password because special care is needed for parameter `password`.

  Among all attributes in the schema file, only one attribute can be specified to have `true` value.

- `linktoentity=""`

  This parameter specifies the entity this attribute links to. In this example, there are three values available for the parameter: `Computer`, `User`, and `Group`, which are defined in tag `<ObjectClassDefinition>`.

  Generally, only attribute `memberof` can be specified to have `group` value, empty value is used for other attributes.

- `multivalued="false"`

  This parameter specifies whether the attribute is single-valued or multi-valued. If you set it to `true`, the attribute is multi-valued; if you set it to `false`, it is single-valued.

- `mustOnResource="false"`

  This parameter specifies whether the attribute is required or optional on resource. Value of this parameter should be set true if this parameter is required on resource. For example, attribute `cn` is required on Active Directory which `mustOnResource` should be set to `true`.

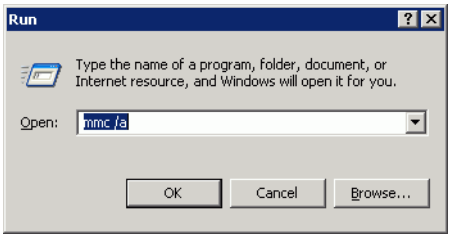
- `name="objectclassuserattributeuserAccountControl"`

  This parameter specifies the attribute name which must be unique in the schema file to connect the tag `<attributeDefinition>` which describes the same attribute in this schema file. It is recommended to form the attribute name by joining together string `objectclass`, string `attributeuser`, and the value of parameter `concero:tafield`.

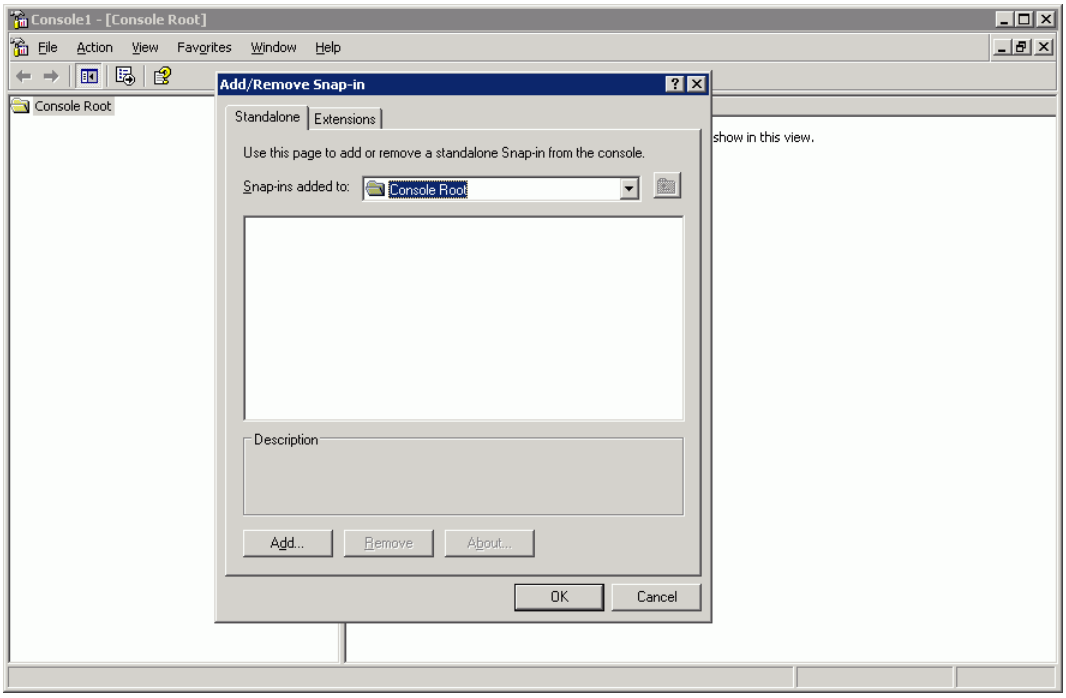
- `objectClass="user"`

  This parameter specifies which objectClass the attribute belongs to. You can obtain the value of `objectClass` from Active Directory Schema by following the steps below:

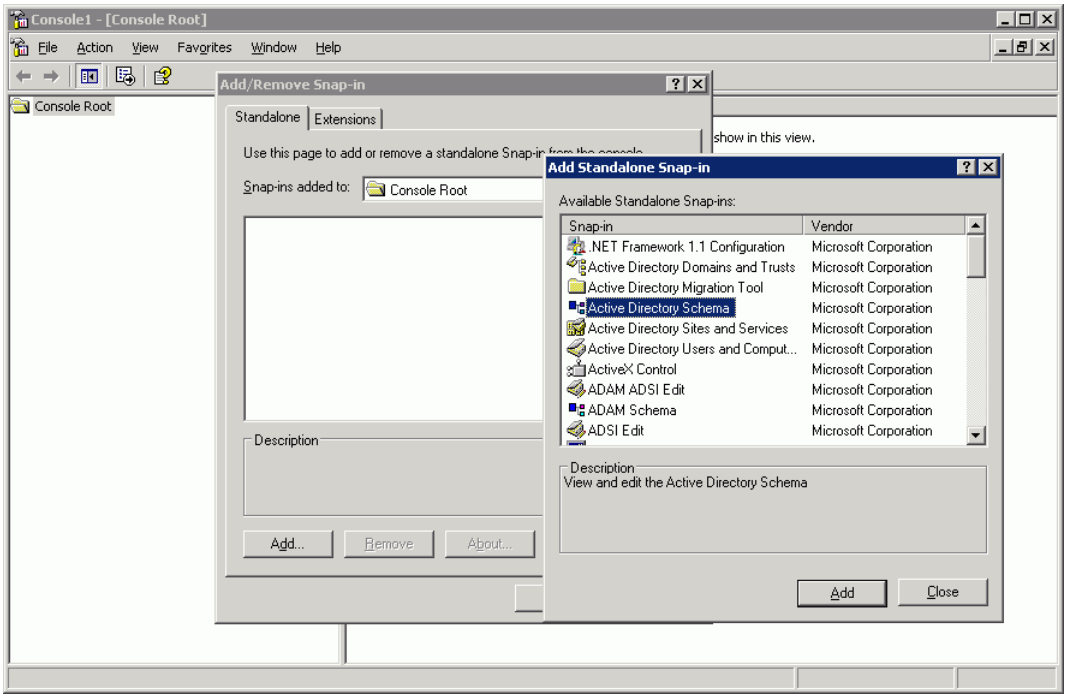
  a. Run **`mmc /a`** on your Active Directory Server:



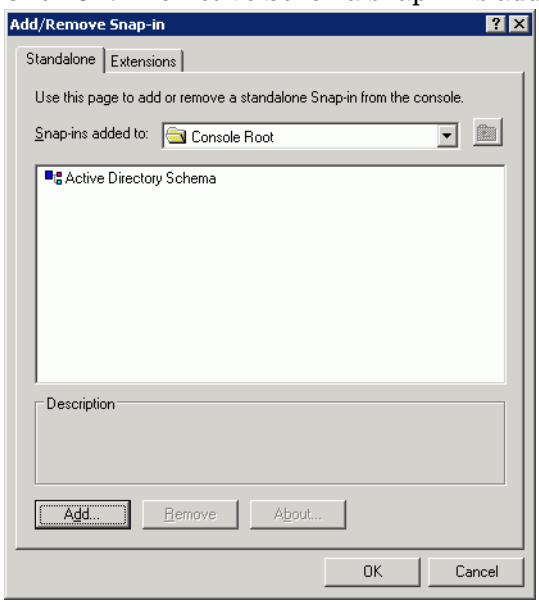
b In the Console window, click **File** → **Add/Remove Snap-in**. The Add/Remove Snap-in windows appears.



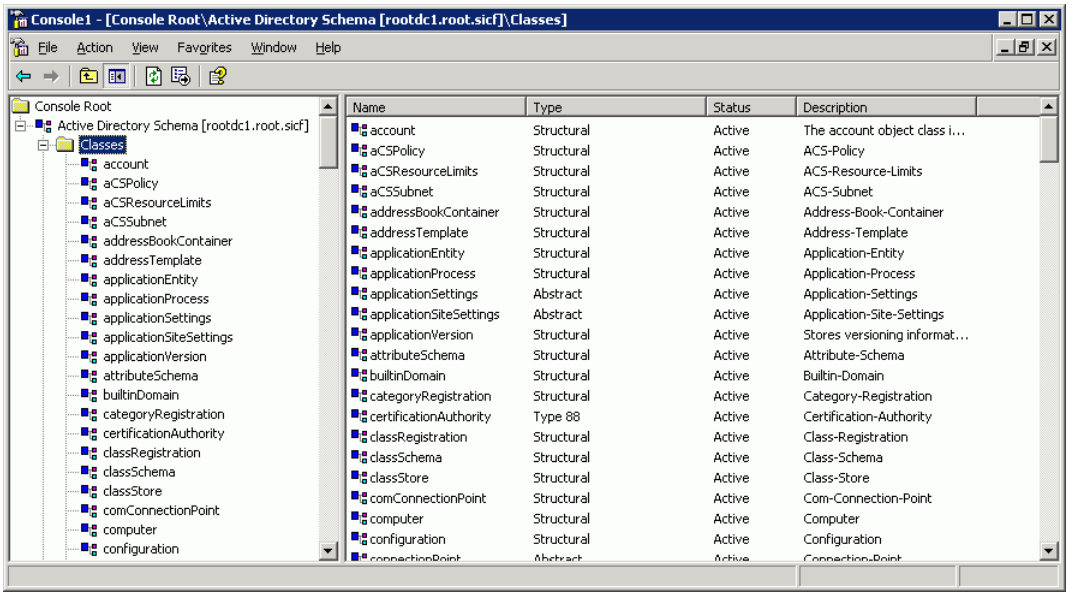
c Click **Add**. The Add Standalone Snap-in windows appears, select Active Directory Schema from the Snap-in list, then click **Add**



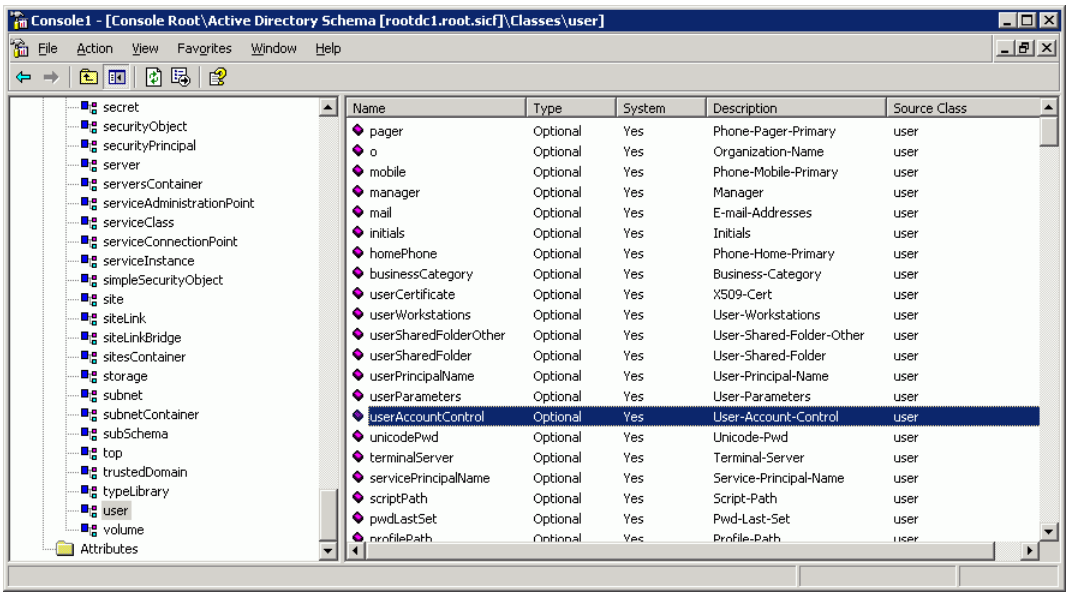
d Click **OK**. The Active Schema snap-in is added.



e In the Active Directory Schema window that appears, expand Classes node in the left panel.



f Scroll down and select **user** class. then find userAccountControl in the right panel. You can find the objectClass of the attribute in Source Class column, for example, the objectClass of userAccountControl is “user” as shown below:



- `objectclasstype="structural"`

  This parameter specifies the objectclasstype of this attribute. There are three values available for objectclasstype in Active Directory:

  — *`Structural`*: Used to instantiate objects (users, servers, and so on) in the directory. This is default value.

  — *`Abstract`*: Provides templates for deriving structural classes.

  — *`Auxiliary`*: Contains predefined lists of attributes that can be included in `structural` and `abstract` classes.

- `ordering=""`

  This parameter is not yet implemented in current version.

- `remexpireValue="0"`

  This parameter specifies whether to remove expiration of the password. If attribute is 0, the password expiration is removed.

- `renameKey="false"`

  This parameter specifies whether the attribute value can be changed.

  Only attribute `cn` can be specified to have `true` value (`renameKey="true"`) in the current version.

- `required="false"`

  This parameter specifies whether the attribute is required in provisioning process on Select Identity. Value of this parameter should be set to `true` if this parameter is required on Select Identity. For example, attribute `sAMAccountName` is required on Select Identity which `required` should be set to `true`.

- `resourceKey="false"`

  This parameter specifies whether the attribute is the resource key that is used to uniquely identify an object on resource.

Only one attribute in the schema file can be specified to have true value for this parameter (`resourceKey="true"`).

- `entityType="user"`

  This parameter specifies the entity for which the attribute can be used. There are three values available for entityType:

  — `user`: the attribute can be used by `user` only

  — `contact`: the attribute can be used by `contact` only

  — `user|contact`: the attribute can be used by both `user` and `contact`.

- `supportedOperations ="UNLINK,LINK,GETATTRIBUTES,GETPARENT,GETCHILDREN,GETALL,RESETPASSWORD,CHANGEPASSWORD,EXPIREPASSWORD,DISABLE,ENABLE,CREATE,DELETE,UPDATE"`

  This parameter specifies the operations the attribute supports on resource.

  The parameter value here is the default value.

- `transform="NO"`

  This parameter specifies whether the type of the attribute can be transformed into another one. It is not yet implemented in the current version.

- `type="java.lang.String"`

  This parameter specifies the attribute type on resource.

Below is an example of tag `<attributeDefinition>`, which describes parameters of the attribute:

```
- <attributeDefinition description="Group_objectclassgroupattributemember"
    name="Group_objectclassgroupattributemember" type="java.lang.String">
  - <properties>
    - <attr name="minLength">
        <value>0</value>
      </attr>
    - <attr name="maxLength">
        <value>255</value>
      </attr>
    - <attr name="defaultValue">
        <value />
      </attr>
    - <attr name="pattern">
      - <value>
          <![CDATA[ [a-zA-Z0-9@]+  ]]>
        </value>
      </attr>
    </properties>
  </attributeDefinition>
```

You can modify the attribute details according to their descriptions:

- `description`: description of the attribute.
- `name`: name of the attribute.
- `type`: type of the attribute.

  ➤ Make sure that the values for `name` and `type` are the same as those in parameters `name` and `type` in tag `<attributeDefinitonReference>`.

- `minLength`: minimum length of the attribute, "0" is used as default value.

- `maxLength`: maximum length of the attribute, "255" is used as default value.
- `defaultValue`: default value of the attribute, empty string is used as default value.
- `pattern`: the pattern to check format of attribute values, `"![CDATA[[a-zA-Z0-9@]+]]"` is used as default value.

Modifying Existing Attribute Mapping

To modify an attribute in the schema file, make sure to modify these two tags in the schema file:

- `<attributeDefinitionReference>`
- `<attributeDefinition>`

Deleting Existing Attribute Mapping

To delete an attribute from the schema file, make sure to delete these two tags from the schema file:

- `<attributeDefinitionReference>`
- `<attributeDefinition>`

Customizing Enable/Disable Mapping

Tag `<concero:objectStatus name="enableUser">` and tag `<concero:objectStatus name="disableUser">` in `ActiveDir.xml` define the attribute and its value used in `enable` and `disable` user, as shown in the screenshot below:

```
+ <attributeDefinition description="objectclassldapv3ConnectorattributeDn" name="objectclassldapv3ConnectorattributeDn"
    type="java.lang.String">
+ <attributeDefinition description="objectclassorganizationalPersonattributepostOfficeBox"
    name="objectclassorganizationalPersonattributepostOfficeBox" type="java.lang.String">
+ <attributeDefinition description="objectclassuserattributepwdLastSet" name="objectclassuserattributepwdLastSet"
    type="java.lang.String">
+ <concero:relationshipDefinition>
- <concero:objectStatus name="enableUser">
  - <concero:attributeMap concero:operation="" concero:resfield="userAccountControl" required="false">
      <concero:attrvalue>{512}</concero:attrvalue>
    </concero:attributeMap>
  </concero:objectStatus>
- <concero:objectStatus name="disableUser">
  - <concero:attributeMap concero:operation="" concero:resfield="userAccountControl" required="false">
      <concero:attrvalue>{514}</concero:attrvalue>
    </concero:attributeMap>
  </concero:objectStatus>
</Schema>
```

Below is an example of tag `<concero:objectStatus name="enableUser">` and tag `<concero:objectStatus name="disableUser">`, which describes parameters of the attribute:

```
- <concero:objectStatus name="enableUser">
  - <concero:attributeMap concero:operation="" concero:resfield="userAccountControl"
      required="false">
      <concero:attrvalue>{512}</concero:attrvalue>
    </concero:attributeMap>
  </concero:objectStatus>
- <concero:objectStatus name="disableUser">
  - <concero:attributeMap concero:operation="" concero:resfield="userAccountControl"
      required="false">
      <concero:attrvalue>{514}</concero:attrvalue>
    </concero:attributeMap>
  </concero:objectStatus>
```

You can modify the attribute name and its value according to their descriptions if you want to customize the operations when user is enabled or disabled:

- `concero:resfield`: the attribute used to indicate user status in `enable` and `disable` user. Only attribute `userAccountControl` is supported in current version
- `required`: whether the attribute is required on Select Identity, `required` is set to `false` in current version
- `concero:attvalue`: the value in Active Directory which represents status of user, for example:

  If user is enabled, the value of attribute `userAccountControl` is 512;

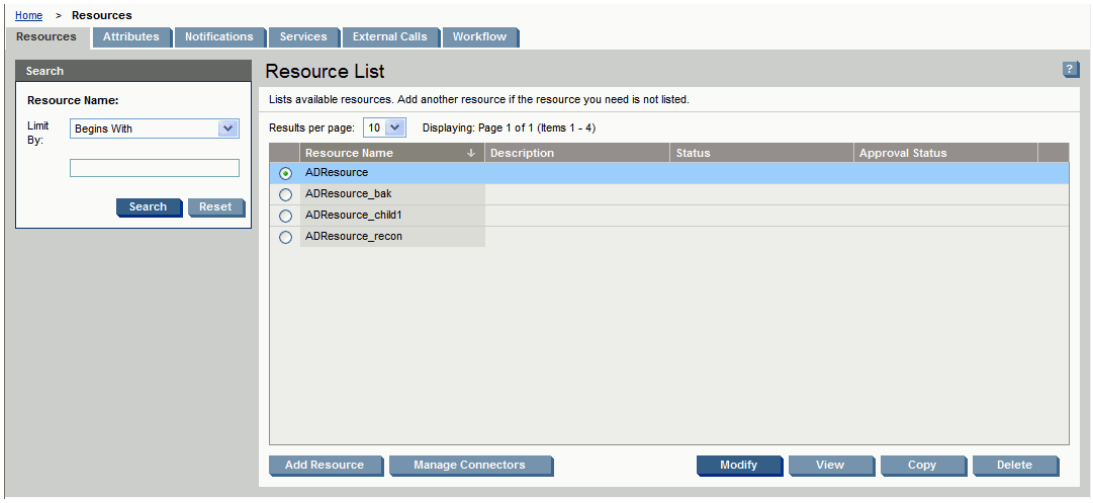
  If user is disabled, the value of attribute `userAccountControl` is 514.

Verifying Attribute Addition/Deletion on Select Identity

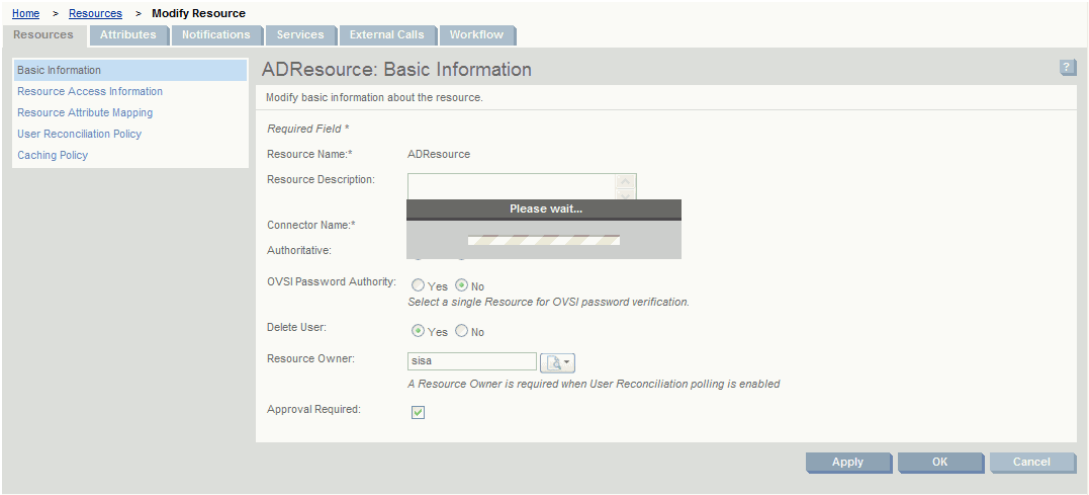
Perform steps below to verify if an attribute is added or deleted on Select Identity:

1. In Select Identity, click **Resources** in Service Studio section.

   In the Resources window, select a resource from Resource List that uses the connector. In this instance, ADResource is selected.



2 Click **Modify**. When Basic Information window appears, click **OK**



3 Select the resource again, and click **View**.

When the Basic Information window displays, click **Resource Attribute Mapping** from the left panel.

Verify in the right window if an attribute is already added/deleted. In this instance, attribute `userAccountControl` is added.