CA Single Sign-On

Implementation Guide

r12.0

This documentation and any related computer software help programs (hereinafter referred to as the “Documentation”) is for the end user’s informational purposes only and is subject to change or withdrawal by CA at any time.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in the Documentation is governed by the end user’s applicable license agreement.

The manufacturer of this Documentation is CA.

CA Product References

This document references the following CA products:

- CA® Single Sign-On CA SSO
- CA® Access Control
- CA® ACF2
- CA® Audit
- CA® Directory
- CA® Top Secret
- Unicenter® Software Delivery

Contact CA

Contact Technical Support

For your convenience, CA provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA product documentation, complete our short customer survey, which is also available on the CA support website, found at http://ca.com/support.

Contents

Chapter 13: Communication Modes 389

Chapter 14: Implementing Citrix Application Migration 399

Chapter 15: Implementing Password Synchronization Agents 415

Chapter 16: Scheduling Maintenance Tasks 429

Chapter 17: Upgrading 447

Chapter 1: Understanding CA SSO

This chapter introduces you to the major components of CA SSO. It gives you an overview of the product and helps you understand how the pieces fit together and how the basic processes work. This chapter is especially useful to people who are new to the product.

This section contains the following topics:

SSO Building Blocks

This section gives you an overview of each of the components that fit together to give you CA SSO functionality. We recommend that you read this section first as it gives you valuable overview information.

CA SSO Server

The CA SSO Server is the heart of CA SSO. The CA SSO Server embeds two data stores:

- CA Access Control, which is a database that stores all administrator, access control and policy information
- CA Directory, which is an LDAP data store where you can store your user data

  You can also configure the CA SSO Server to connect to a third-party data store, such as an Active Directory Service.

The CA SSO Server performs the following functions:

- Builds the application lists for users and sends them to the CA SSO Client
- Retrieves the logon variables which are the user-specific logon data for each application
- Stores SSO scripts and sends them to the CA SSO Client when needed
- Manages data such as access rules in the CA Access Control data store

- Manages data such as tokens in the CA Directory data store
- Connects to data on Active Directory (if you want to use ADS as your user data store)
- Provides authentication (if you choose to use SSO Authentication)

Where it is installed

Install this component first. Always install the CA SSO Server in a server farm configuration with at least two servers in every farm.

How it is installed

Install the CA SSO Server directly from the product DVD.

How it is controlled

Use the Policy Manager, which is an administration interface, to manage the CA SSO Server. In addition to this you can control the:

- CA SSO Server using pslang command line functions
- Users and policies in CA Access Control database using selang command line language

Policy Manager

The Policy Manager is a Windows-based management interface for managing the CA SSO Server and the data stores.

Use the Policy Manager to:

- Configure and connect all SSO components
- Establish access policies
- Create and manage user resources
- Define SSO-enabled applications
- Create session profiles
- Create password policies

Where it is installed

Install the Policy Manager on all computers that your administrators use to control the CA SSO Server.

How it is installed

Install the Policy Manager from the Product Explorer or using a silent installation.

More information:

Implementing the Policy Manager (see page 117)

Authentication Agent

An agent is a program that performs information gathering or processing tasks, and typically interfaces with another software component. An authentication agent forms a link between the CA SSO system and the authentication software. CA SSO comes with several ready-made authentication agents that work with native and third-party authentication methods.

Every authentication method requires a corresponding authentication agent to relay information between the CA SSO system and the authentication software.

Decisions you need to make

You need to choose which authentication method or methods you want to implement. This usually depends on what authentication you currently use.

Authentication Agents Provided with CA SSO

CA SSO comes with a native authentication method that you can use immediately out-of-the-box. The CA SSO authentication agent is installed automatically with the CA SSO Server. Native SSO authentication uses username and password to authenticate users.

Note: We do not recommend that you use the SSO authentication method in a production environment if you intend to use Microsoft Active Directory as your user data store.

Where it is installed

The location of the authentication agent depends on the method of authentication (authentication software) and the level of security you require.

The server where the authentication agent resides is called the authentication host. The corresponding authentication software is usually also located on the authentication host, but can be located on another computer for security reasons.

You can configure your system in different ways and install your authentication agent in different places. The following table shows where you might typically install the authentication agent for the different authentication methods. It is rare that you need the authentication agent in more than one location.

Authentication Method	Authentication Agent Location
SSO	CA SSO Server
Windows	ADS Domain Controller
LDAP	Separate authentication computer, usually the LDAP directory server If you want to use Active Directory enhancements, the LDAP authentication agent must be installed on a Windows machine on the appropriate domain.
Certificate	Separate authentication computer, usually the Certificate Authority
RSA SecurID	Separate authentication computer, usually the RSA server

How it is installed

Install the authentication agent from the Product Explorer or using a silent installation.

If you plan to use third-party software, it must be already installed at the site before CA SSO primary authentication agents are installed. Authentication hosts have to be defined in the CA SSO Servers using the Policy Manager.

Your CA representative can help you with your specific authentication requirements.

More information:

Implementing Authentication (see page 241)

User Data Store

The CA SSO Server installs CA Directory as the user data store. You can configure the CA SSO Server to use a different LDAP data store such as Microsoft Active Directory after installation if you already use this in your organization.

The user data store holds information about:

- Users and user groups
- Logon information

You can populate this data store with information from existing data stores in your organization, during or after product installation. You can import information by running a utility, or by using the command line interface.

Where it is installed

CA Directory is installed on the CA SSO Server computer.

How it is installed

CA Directory is installed automatically when you install the CA SSO Server.

How it is controlled

You can control CA Directory using either the Policy Manager or one of the CA Directory management tools such as JXplorer.

Administrative Data Store

The CA SSO Server installs CA Access Control as the embedded administrative data store.

The Administrative data store holds information about:

- Administrators
- Applications
- Access rules
- Password policies
- Session profiles
- Resources
- Configuration Parameters

You can populate this database with resource and application information from existing data stores in your organization, during or after product installation. You can import information by running a utility, or by using the command line interface.

Where it is installed

CA Access Control is installed on the CA SSO Server computer.

How it is installed

CA Access Control is installed automatically when you install the CA SSO Server.

How it is controlled

You can control CA Access Control using the Policy Manager or selang command line language.

CA SSO Client

The CA SSO Client is an application that lets users work with CA SSO. This is the only CA SSO component that end users see.

The CA SSO Client software:

- Lets users enter their credentials
- Verifies the users' credentials by communicating with authentication agents
- Retrieves users' application lists and logon details from the CA SSO Server
- Displays users' applications to the user
- Downloads and executes SSO scripts to automate processes such as logging on for the user

Where it is installed

Install the CA SSO Client on every end-user workstation. The only exception to this is some thin-client environments, where CA SSO is only used to facilitate web access.

How it is installed

Install the CA SSO Client using the CA SSO Product Explorer from the product DVD. This is very straightforward, but can be time consuming if you have to install the CA SSO Client on a large numbers of user desktops. Alternatively, you can roll the CA SSO Client out to a large number of end user machines on a network using a silent installation in combination with a software distribution tool.

How it is controlled

The CA SSO Client behavior is controlled by the Client.ini file and the Auth.ini file. You must install the CA SSO Client at least once, using the Product Explorer to get a copy of the INI files. You can then customize the INI files and distribute them so that when you roll the CA SSO Client to a large number of users, using the silent installation, the CA SSO Client is already customized.

Decisions you need to make

You must plan what functionality you want from the CA SSO Client and what you want your users to experience from the CA SSO system. Decisions you need to make, include:

- What method of authentication are you planning to implement
- How you want users to access the SSO system and SSO-supported applications
- Whether you want offline operation
- Whether you want shared workstation functionality
- Whether you want session migration (Citrix Metaframe environments only)
- Whether you want to limit user sessions

How users access their applications

Users can access their SSO-enabled applications in a number of different ways including:

- From the Launchbar interface
- From the SSO Tools interface
- As menu items in a Windows Program Group
- As shortcuts on their Windows desktop
- From the SSO Status Icon in the Windows taskbar

More information:

Implementing the CA SSO Client (see page 125)

SSO GINA

When a user logs onto a Windows computer, they enter their credentials using the Microsoft GINA. The GINA (Graphical Identification and Authentication library) is the component of Windows that provides secure authentication and interactive logon services. You can replace the Microsoft GINA with the CA SSO GINA.

When to deploy it

Benefits of using the SSO GINA include:

- One-step authentication to both the workstation and to CA SSO
- Using any of the SSO-supported authentication methods for Windows logon which enhances security
- Shared computer mode functionality, if required

How to install it

Install the SSO GINA with the CA SSO Client. When the CA SSO Client is installed with the SSO GINA, it replaces the Microsoft GINA. If the CA SSO Client is uninstalled, the SSO GINA is likewise uninstalled, and the Microsoft GINA is reinstated.

After you install the SSO GINA, you must configure it using the Client.ini file.

How it is controlled

The SSO GINA behavior is controlled by the Client.ini file.

The following are the INI entries related to SSO GINA in the Client.ini file:

```
[GINA]
    LogonBitmap=
    LogonTitle=
    LogonText=
    LockedBitmap=
    LockedTitle=
    LockedText=
    Font=
    FontSize=
    GinaPassThrough=
    LogonCAD=
    FetchDomainsFromSystem=
    Domains=

[GINA/SystemLogon]
    NetWareLogon=
    NetWareServer=

[GINA/StationLock]
    EnableOsUnlock=
    DisableShutdown=
    DisableLogoff=
    UnlockStationMode=
    ShowLockedUsername=
    EnableSSOLogoff=
```

What the SSO GINA looks like

The SSO GINA has four dialogs that the user sees according to their actions and the state of the workstation, these are:

- Welcome
- Authentication
- Security
- Locked

Limitations of the SSO GINA with Terminal Services

The SSO GINA only supports a subset of terminal services functionality: the SSO GINA supports Remote Administration Mode but does not support Application Server Mode.

The SSO GINA with Remote Administration Mode lets administrators gain access to a workstation using an RDP (Remote desktop) client. We recommend that when an administrator connects to a computer:

- it must have no prior logons (this usually means that the computer must be rebooted before the remote administration logon attempt)
- the administrator must select Windows Logon Only instead of using any other SSO authentication method

Standard Operating Systems for GINA

SSO GINA is only supported on Windows XP, 2000, and 2003 platforms. For Windows Vista you must deploy the SSO Credential Provider.

More information:

Implementing the CA SSO Client (see page 125)

SSO Credential Provider

When a user logs onto a Windows Vista computer, they enter their credentials using the Microsoft Credential Provider. The Microsoft Credential Provider is the component of Windows Vista that provides secure authentication and interactive logon services. You can replace the Microsoft Credential Provider with the CA SSO Credential Provider.

When to deploy it

Benefits of using the CA SSO Credential Provider include:

- One-step authentication to both the workstation and to CA SSO
- Using any of the CA SSO-supported authentication methods for Windows logon which enhances security
- Shared computer mode functionality, if required

How to install it

Install the CA SSO Credential Provider with the CA SSO Client. When the CA SSO Client is installed with the CA SSO Credential Provider, filters out the Microsoft Credential Provider by default, and provides the CA SSO Microsoft Credential Provider wrapper for logging onto the Vista machine without logging onto SSO. This is similar to the GINA pass through feature.

How it is controlled

The CA SSO Credential Provider behavior is controlled by the Client.ini file.

The following are the INI entries related to Credential Providers in the Client.ini file:

```
[CredentialProvider]
    EnableSSOCredentialProvider=
    AddLocalMachineToDomainSelectionOptions=
    FetchDomainsFromSystem=
    FilterMSProvider=
    Domains=
    LoginBitmap=

[CredentialProviderTileImages]
    ServerSetTile=
    MSCPTile=
    SSOCPTile=
    WINCPTile=
    LDAPCPTile=
    RSACPTile=
    CERTCPTile=

[GINA/StationLock]
    UnlockStationMode=
```

What the CA SSO Credential Provider looks like

The CA SSO Credential Provider has four dialogs that the user sees according to their actions and the state of the workstation, these are:

- Welcome
- Authentication
- Security
- Locked

Limitations of CA SSO Credential Providers with Terminal Services

The CA SSO Credential Provider supports only a subset of terminal services functionality. It supports remote administration mode only, it does not support application server mode.

In the remote administration mode, the CA SSO credential provider lets an administrator login to a computer using a remote desktop client with the following limitations:

- The remote computer must be restarted before an administrator logs in using the remote desktop client.
- The administrator must only use the Windows Only Logon. The administrator must not use any of the authentication methods to logon.

Standard Operating Systems for CA SSO Credential Provider

CA SSO Credential Provider is supported only on Windows Vista.

SSO Scripts

A CA SSO script is a script written in a special extended version of a command language called Tcl. The CA SSO Client runs a Tcl script and performs a task, or series of tasks, for the user. Scripts can be used for a wide variety of tasks. A logon script, for example, is written to automatically log a user in to an application (automatically insert the correct user's name and password in the relevant fields of the logon screens). Scripts can also be written and configured to be executed when a specific event occurs, such as CA SSO Signoff or a Windows lock event.

There is some overhead to create these scripts up front, but they provide excellent flexibility and let you write scripts to automate almost any task that a user could perform. You should think about tasks that users perform that could be automated.

CA SSO scripts are written in a special extended version of the Tcl scripting language that gives you the use of variables, conditions, loops, procedures, and other common programming devices with a minimum of complexity. Prior experience with Tcl is not required to be able to write these, but some programming experience is an advantage. The security or system administrator in charge of CA SSO is responsible for preparing the logon scripts. These scripts are written during implementation and typically do not affect the day-to-day administration of CA SSO.

You can also use the CA SSO Application Wizard to add your applications to CA SSO without the need to learn Tcl scripting for basic scripting tasks.

Where SSO scripts are stored

You need a script for every application you want users to launch through CA SSO. These scripts are stored on the CA SSO Server in the scripts directory. You can change the directory by specifying a custom value in the Windows Registry.

How SSO scripts are controlled

Scripts are run using the SSO Interpreter, which is automatically installed with the CA SSO Client. Use the Policy Manager to assign a script to a particular application.

What information SSO scripts require

CA SSO scripts use bits of information called script variables (also called logon variables). This is the information retrieved from the CA SSOServer and inserted into CA SSO scripts for a specific user. For example, when CA SSO executes a logon script, the CA SSO script retrieves the username and password for that application and "types" them into the relevant fields on the application's logon window.

For CA SSO to execute a logon script, the SSO Client must have the username and password to insert into the application logon screen. In this instance, the CA SSO script retrieves the logon variables which must include the username and password for that user.

SSO scripts can also make use of other variables such as Windows system variables, and these are created based on their value on the end-user computer. For more information, see the *Tcl Scripting Reference Guide.*

More information:

Adding Applications (see page 309)

SSO-Enabled Applications

An SSO-enabled application is any application you want users to access using CA SSO. SSO-enabled applications can be Windows, mainframe, or web-based applications that you want your users to have access to after they have authenticated. The SSO-enabled applications can be located on the user's computer or on a computer connected to the network. Every SSO-enabled application must have its own SSO script.

How SSO-enabled applications are controlled and defined

SSO-enabled applications are defined on the CA SSO Server using the Policy Manager.

How applications are organized for the user

Every user has an application list generated for them. This list contains all of the SSO-enabled applications that the user is authorized to use. If the user is a member of a group, all of the applications that the group can access appear on the user's SSO application list. The CA SSO administrator or implementation team can customize how users access their application lists.

How users access their SSO-enabled applications

For an application to appear in a user's CA SSO application list, the administrator must create an application and assign the application to the user or user group.

After users have been authenticated they can access their SSO-enabled applications in any of the following ways:

- SSO Launchbar
- Status Icon, Applications menu
- SSO Tools
- Windows Start menu, Programs, SSO Programs
- Shortcuts on the Windows desktop
- Windows Startup menu, which means that the application starts automatically when the user logs on

You can limit which methods users can use to access their SSO-enabled applications. You can also change the icon and caption associated with an application. You may choose to do this to help users identify familiar applications.

Session Administrator

The Session Administrator is a web-based interface for managing user sessions. You only need to use this when you are managing sessions in your CA SSO environment. You can create automatic session profiles using the Policy Manager.

Where it is installed

Install the Session Administrator on a web server on the network.

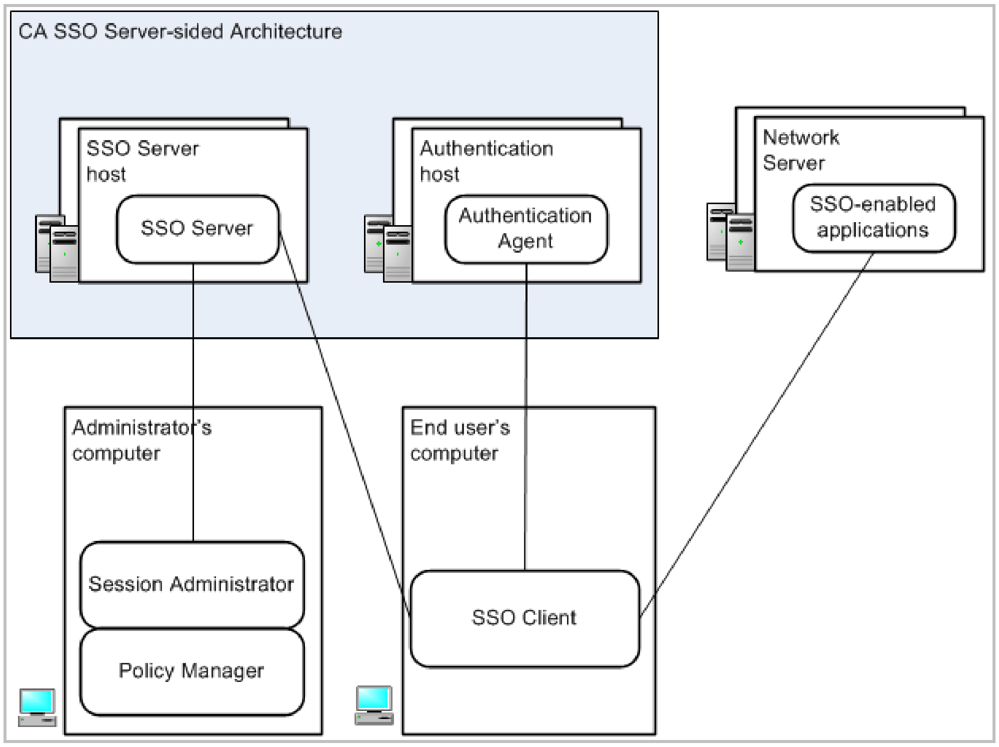
How it is installed

Install the Session Administrator from the Product Explorer or using a silent installation.

Architectural Overview

This diagram shows you a typical installation of CA SSO and shows the relationships between the components. There are many ways to install the configure CA SSO, but this represents a typical installation.

You might want to start reading about each of the components in this chapter and refer back to this diagram as you go so you can see how they all fit together.



Implementation Overview

In many cases, the most efficient implementation strategy is a sequential process. Here are the suggested implementation steps in order of components.

1. Install the CA SSO Server
2. Install the Policy Manager (on administrator workstations)
3. Populate or configure the data stores

4. Install the authentication agent(s)
5. Write the logon scripts (and other scripts)
6. Install the CA SSO Client (on end-user workstations)
7. Install the Session Administrator (optional)
8. Install the Password Sync Agent (optional)

You may want to start development on step 5 (Write the Logon Scripts) early, in parallel with the other steps, to make sure they are ready in due time.

After each installation and configuration step, we strongly recommend that you verify that the component added is working as expected. For example, after performing step 3, use the Policy Manager to perform an ad-hoc verification that user and application data is assigned as expected.

Note: All operating system clocks must produce a reliable and correct timestamp for the time-zone where each machine hosting any SSO components are located. For example, a machine located in New York hosting an CA SSO Server must have its OS clock set to US Eastern Daylight Time (EDT) while a machine located in San Francisco hosting an LDAP Auth Agent must have its OS clock set to US Pacific Daylight Time (PDT).

CA SSO Functionality

This section is designed to help you choose which bits of CA SSO functionality you want to deploy.

Offline Operation

Offline operation is the ability of the CA SSO Client to continue to work even when it cannot establish a connection with the CA SSO Server or the authentication agent. When a user uses CA SSO in offline mode, they can access specific applications that have been marked for offline use.

When to deploy offline operation

You may choose to deploy offline operation for two main reasons.

Ensuring uninterrupted service

You want your users to have uninterrupted access to SSO, even if the connection to the network periodically goes down. In this situation they can log onto SSO and launch all their applications that you have marked for offline use, regardless of network connection.

Assisting laptop users

You want your laptop/remote users to be able to use SSO when they are not connected to the network. This includes their ability to log on to Windows using the SSO GINA and their ability to access SSO-enabled applications. This is useful if you want users to connect to a remote network from home. You can set up a VPN Client as a SSO-enabled application that can be launched offline, which then connects the user to the network.

Constraints of offline operation

Here are the constraints you need to think about when you deploy offline operation.

Limited authentication methods

During offline operation when the CA SSO Client can only authenticate users with one of the following methods:

- Certificate
- LDAP
- SSO
- Windows

This means that you may have to allocate one of these methods of authentication to your users if you currently deploy a different method of authentication. This also means that a user can authenticate using one of these methods even if the CA SSO Client cannot connect to the network.

Shared computer constraint

Users can only access offline functionality on a computer that they have previously logged on to in full online mode. To enable offline operation the CA SSO Client must cache the user's logon details so that these can be used when it is offline. To cache these details the user must successfully authenticate on that computer. This may be inconvenient for users that need to log onto multiple computers in different locations because you cannot guarantee that they return to the same computer when the system goes offline.

Offline Operation and Session Management

Offline operation is incompatible with Session Management because the CA SSO Server cannot control the number of offline sessions a user may have open while they are offline.

Offline Operation with the SSO GINA and SSO Credential Provider

If users logon using the SSO GINA or SSO Credential Provider, they can only log on if you have marked their domain application for offline use. Again, in this case, for the logon to be successful, the logon credentials for the user must have been cached during a previous successful windows logon. Otherwise, the user cannot login, even when the domain application is marked as offline.

Application Launchpad Using CA SSO

You can even add applications that have no password protection to CA SSO so you can set up CA SSO as a type of launchpad for each user that includes all applications you think a user might need, not just their secure applications.

To add an application to CA SSO, create a script for that application. Scripts are extremely versatile and you can create scripts that automate many end-user tasks, not just logging on to applications. You must plan what tasks you want CA SSO to automate.

Shared Workstations

You can configure the CA SSO Client to suit how people use their computers. For example, you might have a kiosk-style workstation environment where lots of people access a single computer each day. Or you might have one computer per person.

These computer modes affect how the computer is unlocked and how the users of that machine access their SSO-enabled applications and the Windows desktop. You must make sure you understand how users access computers in your organization before you decide which mode to work with.

The computer modes are:

Single-user workstation mode

This is used in non-shared workstation environment. The computer can only be unlocked by the person who locked it (or a systems administrator). This therefore suits a situation where the same person uses this computer all the time. This option provides the greatest security.

Scenario

Nancy sits at one workstation full-time. She does not share her workstation with anyone else. She is the only person who logs into the domain and uses CA SSO from this computer.

Fully Shared Workstation/Mode 3 (kiosk)

This is used in a full shared computer environment. The computer can be unlocked by any SSO user, but there is no reference to the underlying Windows user. This suits a situation where two or more people share a computer and Windows setup, but each user wants to have their own customized CA SSO applications. This is like the semi-shared workstation mode 1 option, but is much faster and suits an environment where several people may have to use one computer in quick succession.

In Vista, you can configure the credential provider to log in to the Windows desktop automatically upon startup using the Windows username and password stored in the registry keys for Credential Provider. So, users share the same underlying Windows credentials but different CA SSO credentials.

Scenario

A busy ward in a hospital has a computer and printer for general use. This computer is used by many doctors who need to access data quickly and print things out without going back to their offices. This computer has a generic Windows desktop and settings that connect to the printer. Each doctor can unlock the computer and see their CA SSO applications. From here they can print to the printer right beside the computer because this Lock Option uses the local settings of the shared Windows setup.

Semi-shared workstation/Mode 1 (Single Windows desktop for all users)

This mode caters to multiple users on one computer who can share a single Windows desktop. This is used in a semi-shared computer environment. The computer can be unlocked by any SSO user, as long as that user shares the underlying Windows logon. This suits a situation where two or more people share a customized Windows setup, but need access to their own CA SSO applications on a workstation. All users work as different CA SSO users using the same Windows profile.

You must write a logoff script for each user. When either user is logged off, their logoff script runs, closing all of their open applications.

Scenario

Hillary and Mike both work in Human Resources and spend a lot of their time in interviews, so they share one workstation. They share a Windows desktop that shows the applications that relate to their job, but they need to have separate access to their own CA SSO applications. When either of then unlocks the workstation in their own name they can see the same Windows setup, but their own specific CA SSO applications.

Semi-shared workstation/Mode 2 (Unique Windows desktops for each user)

This mode caters to multiple users on one computer who all need their own Windows Desktop.

This is used in a semi-shared computer environment. The computer can be unlocked by any SSO user, but if the new user has a different underlying Windows logon, the old Windows user is logged out and the new user is logged on. This suits a situation where two or more people share a computer and each user wants to have their own customized Windows setup as well as their own CA SSO applications. This method is slower, because it completely logs one user off Windows and then logs the next user on and is not recommended.

You must write a logoff script for each user. When either user is logged off, their logoff script runs, closing their open applications.

Scenario

Peter and Sally both share a workstation. They do very different jobs so they each want their own Windows desktop and their own CA SSO sessions. Peter works in the morning and leaves at midday. When Sally starts work in the afternoon she unlocks the workstation in her own name and sees her own Windows desktop and her own CA SSO applications.

Single user Windows session/Mode 4 (Single CA SSO user per Windows session)

This mode is used in a non-shared Windows session environment (one CA SSO user for one Windows user, but multiple Windows sessions (accounts) are allowed).

A Windows session corresponds to exactly one Windows account. The Windows account is not shared by multiple SSO users. The locked Windows session can only be unlocked with the credentials that match those of the last SSO user. If the SSO user that unlocks the station has different Windows credentials, a new Windows session is opened without logging the previous Windows user off. If this session is locked too, each of the signed-on SSO users can only unlock the Windows session associated with it. Other SSO users can log in, only if they have different Windows users. (A new Windows session is created.) This means that multiple users share the same machine resources.

Scenario

Sharon logged into her Windows session to update a patient's file and then she locked her Windows session. George, who has different underlying Windows credentials logged onto the machine, using his credentials, which gave him access to his new Windows session and has now locked his Windows session. Both users can interchangeably use the machine just by unlocking their corresponding sessions without seeing each others applications.

Multiple users Windows session/Mode 5 (Multiple SSO users per Windows session)

This mode is used when two or more people share the same Windows account but need access to their own SSO applications on a computer (multiple SSO users for one Windows user, but multiple sessions are allowed).

The locked Windows session can be unlocked by any SSO user as long as that user has the same underlying Windows user. If the unlocking SSO user has a different underlying Windows user, a new Windows session is opened without logging off the previous Windows user. This means that multiple users share the same machine resources.

Scenario

Jack and Roger share the same Windows account, but each needs to access their own SSO applications on a computer. Initially, Jack logs into the Windows account using SSO Credential Provider and locks the Windows session. Roger chooses to login using SSO Credential Provider and logs in to the same Windows session that Jack was logged in to. By this time, Jack is logged off from SSO and Roger logs in. Roger cannot see any of Jack's applications because the logoff scripts configured for Jack run immediately after Jack logs off SSO, thereby closing all of Jack's applications. If Jarrod logs on using different Windows credentials, a new Windows session is opened.

Note: Modes 4 and 5 are only supported on Microsoft Vista platforms with SSO Credential Provider installed.

Session Control

A session is the period of time a user is logged onto the CA SSO Client. You can set rules that limit the number of sessions a user has open at once and how those sessions behave and when they expire. You can also control sessions manually using the Session Administrator.

You can use Session Management to:

- Limit the maximum number of sessions a user can have open simultaneously
- Define what happens when a user attempts to exceed this number of sessions
- Set an expiration time for sessions
- Manually terminate any session

You can also install the Session Administrator, if you want to control individual user sessions manually instead of using a session profile.

Central CA SSO Client Configuration

The behavior of the CA SSO Client is controlled by the CA SSO Client configuration files (Client.ini and Auth.ini). These files are stored on each CA SSO Client computer and control the behavior of the CA SSO Client that computer. You can configure these files to automatically check a central server for new configuration files using a setting in the Client.ini file.

The CA SSO Client can detect an updated ini file on the central server, pull it down and apply the changes ‘in flight’. This means that you do not need to restart the CA SSO Client if making changes this way to apply them.

Password Controls

Password control and protection is a primary focus for CA SSO. Here are some of the password controls available:

Password Policies

A password policy is a set of rules that defines password behavior and enforces a certain level of security.

This password policy controls the password that users enter to log onto the CA SSO system. You can create password policies using the Policy Manager.

You can set the following password constraints:

- Minimum and maximum length
- Alphanumeric/upper and lower case requirements
- Upper and lower case combination
- Password change interval
- Password history (how many password changes required before reuse is allowed)
- Grace logons (how many times someone can use a password after it expires)

Password Synchronization between applications

Users can synchronize all their SSO-enabled applications so that they can simplify their passwords if they choose to access the application outside SSO.

Password Synchronization of Network logon

You can choose to implement Password Synchronization between the Windows Active Directory Domain logon and the SSO logon.

This synchronization can be done in both directions: the domain controller can notify SSO of changes the user has made to their Windows password, as well as SSO notifying Windows when changes are made to applications which use the domain credentials.

Script Caching to Reduce Network Traffic

You can reduce network traffic by storing logon scripts in a cache on the CA SSO Client computer. If you enable script caching, each time a user launches an SSO-enabled application the logon script for that application is then stored on the CA SSO Client computer for a set period of time, for example a period of days. Within that time any user on that computer who launches that application invokes the cached logon script instead of contacting the CA SSO Server and downloading the logon script each time.

This functionality is separate from offline operation. Any application marked for offline operation automatically has its logon script cached, regardless of whether you enable script caching.

Note: Script caching does not store any private information such as logon credentials.

More information:

About the CA SSO Client (see page 125)

Task Automation

CA SSO scripts can do more than simply launch applications and enter user credentials. Because the Tcl scripting language is so flexible, you can create scripts to do many things.

Scripts can also:

- Close or log off applications
- Change and synchronize passwords
- Automate repetitive tasks
- Automate long navigation trails

Scenario

Ken, a busy doctor, must access an application that permits him to enter patient data. Each time Ken logs in to this application, he must navigate through four windows and enter default data before reaching the screen he actually needs. An administrator can write an SSO script to automate this process and increase Ken's productivity.

For more information, see the *Tcl Scripting Reference Guide*.

CA SSO Logon to Windows

You can configure the CA SSO Client so that users can log onto Windows using the SSO GINA instead of the Windows GINA or by using the SSO Credential Provider if you use Windows Vista. This improves security because it lets you authenticate users with any authentication method that is compatible with CA SSO, and removes the need for users to remember their domain password. This password can then be set to a longer, more secure value.

Message of the Day

The system administrator can define a Message of the day (MOTD). MOTDs are a way to communicate with users when they log on to CA SSO. For example, you might use an MOTD to notify users about changes in working procedures. The MOTD is invoked in one of two ways:

Global MOTD

This message is displayed when the CA SSO Client starts. This must be configured on the CA SSO Server.

Application MOTD

This message is displayed when a specific SSO-enabled application starts. This is configured on the CA SSO Server.

Client MOTD

This message is displayed before signon on the Client machine. This is configured on the CA SSO Client.

The text of these messages is defined on the CA SSO Server and is sent to the user's workstation when requested by the CA SSO Client. Each message resides in a separate file, in the motd directory in the CA SSO Server installation area. You can change the directory by specifying a value for Widows servers in the Windows Registry.

The global MOTD resides in a file named motd and each application MOTD resides in a file named motd.appl, where appl is the name of the application in the data stores. These files are optional; if a file by this name is not found, CA SSO does not display any message of the day for the specific application.

Configurable User Interface

When you install the CA SSO Client on an end user computer, you can choose one or more of the CA SSO Client interfaces for your end users. By default users can access all three.

These interfaces are:

- Status Icon
- SSO Launchbar
- SSO Tools

Each interface lets the user:

- Launch their SSO-enabled applications
- Lock the computer
- Log off SSO
- Change their SSO   authentication credentials

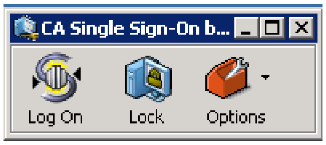
You can choose to configure any of the following options:

- Remove one or more interfaces
- Remove buttons or options such as Logoff
- Change the appearance of the application icons
- Allow the Launchbar to dock to edges of the screen
- Force the Launchbar to always stay on top of other windows

We recommend that you install the CA SSO Client on a test workstation and become familiar with each interface, as well as the Client.ini file which controls the behavior of each interface before you decide on the final functionality and roll this out to a large number of users.

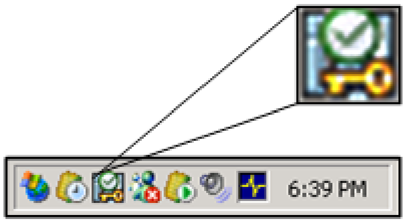
SSO Launchbar

The SSO Launchbar looks like this:

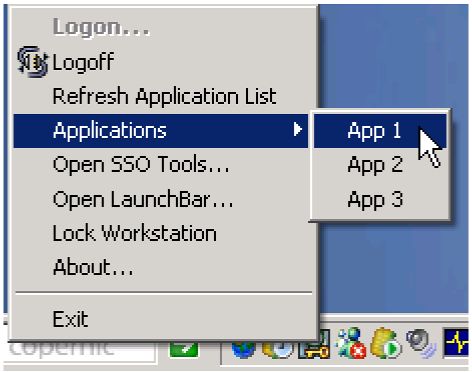


Status Icon

The Status Icon shows the user their SSO status and looks like this:

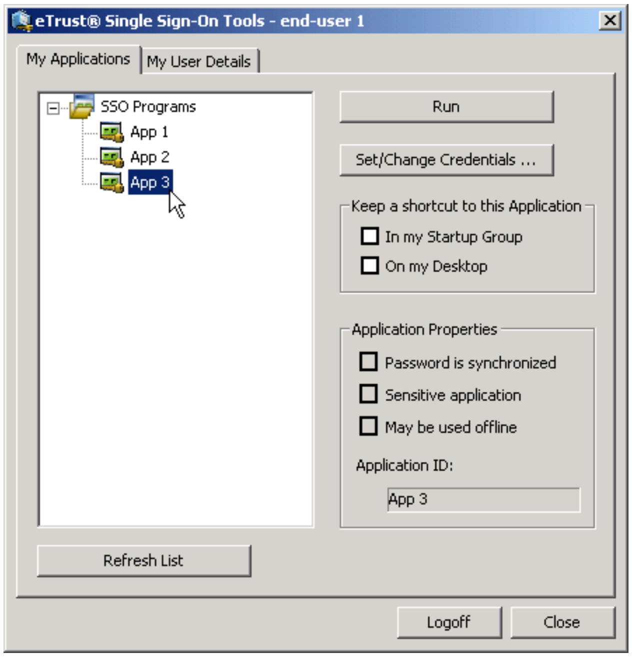


When a user right-clicks this icon, they see this:



SSO Tools

The SSO Tools window looks like this:



Keep a shortcut to this Application

□ In my Startup Group

□ On my Desktop

Application Properties

□ Password is synchronized

□ Sensitive application

□ May be used offline

Application ID: App 3

Note: CA SSO Client and the agents now support Section 508 specifications. Section 508 specifies easy access and use of information and data to users with disabilities comparable to users without disabilities.

Windows Start Menu

Users can launch CA SSO from the Windows Start menu by selecting Start, All Programs, CA, Single Sign-on. You can also add CA SSO and SSO-enabled applications to the Window Startup Folder, so that CA SSO automatically starts when the user logs onto their computer.

After they are onto CA SSO, users can access their SSO-enabled applications from the Windows start menu by selecting, Start, All Programs, CA, Single Sign-On.

Desktop

Users can launch their CA SSO applications from their desktop. They do this by right-clicking on an application on the Launchbar and selecting Advanced, or by selecting an application on SSO Tools and checking keep a Shortcut to this Application on my Desktop. This functionality can be disabled if the administrator prefers not to provide this to their CA SSO users.

Common Processes

This section explains how the following common processes are performed by CA SSO:

- Authenticating users
- Launching applications

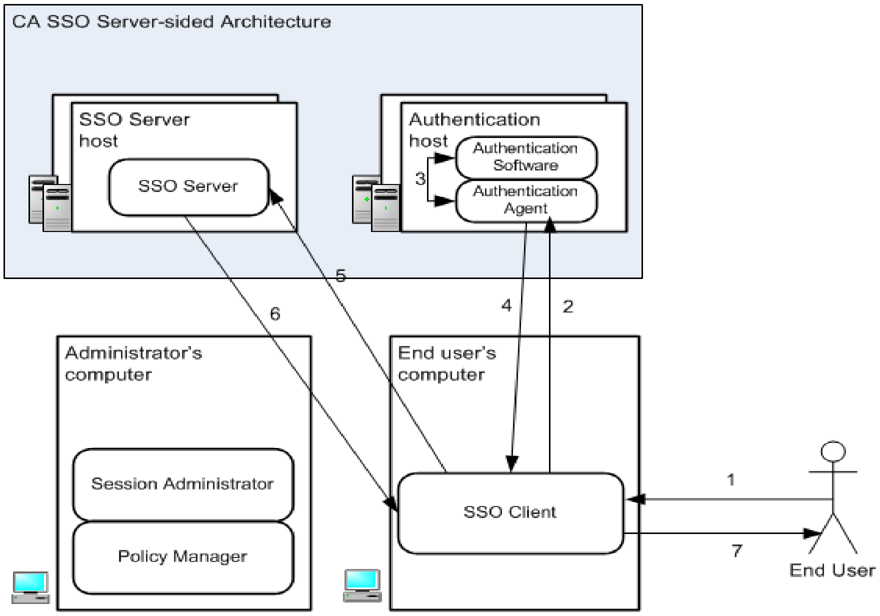
Each description includes a diagram that shows how the component fits into the architecture of CA SSO.

How Authentication Works

Primary authentication is how users identify themselves to the system.

After the user has entered their credentials, the CA SSO Client sends those credentials to the authentication agent. The authentication agent acts as a go-between, it passes those credentials to the relevant authentication software and receives confirmation back. The authentication agent then produces a ticket and sends it back to the CA SSO Client.

The following diagram shows how the authentication process works using a third-party authentication method as an example.



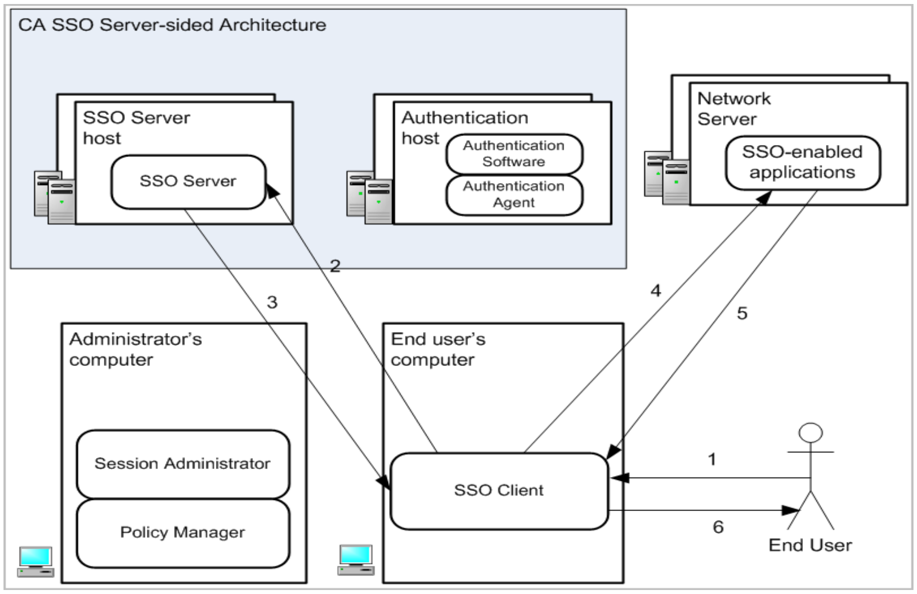
1. The user enters their credentials in to the CA SSO Client.
2. The CA SSO Client sends the credentials to the authentication agent
3. The authentication agent connects to the authentication software, verifies the credentials and creates a ticket.
4. The authentication agent sends the ticket back to the CA SSO Client. This ticket is time stamped and expires after a set period, or persists until the CA SSO session is terminated, depending on the expiration settings on the CA SSO Server.
5. The CA SSO Client uses the ticket to log onto the CA SSO Server to check which applications are allocated to that user.
6. The CA SSO Server sends a list of the applications to the CA SSO Client.
7. The user sees a list of their SSO-enabled applications.

How Applications are Launched

Once end users have been authenticated, they can select and launch any application that has been added to their CA SSO application list.

Each application must have a Tcl script to perform the logon functionality and any other tasks required to help the user.

You can configure CA SSO to let you launch different application types including Windows, Web, and mainframe. This section shows how to launch a Windows application and how to access a Web resource.



Before this process begins, the user has successfully authenticated.

1. The user launches their email application from their CA SSO list.
2. A request to launch the application is sent to the CA SSO Server together with that user’s SSO ticket.
3. The CA SSO Server checks whether the user is permitted to access that application. If they are, the CA SSO Server sends back a script together with the login variables to launch the application.

   In this example, the application is installed on the user’s computer and the script launches the application and enters the relevant username and password.
4. The email application checks for the user's emails on the email server.
5. The emails are downloaded on to the user’s computer.

   The user is now logged in and able to access their email.

Launch Applications on Vista

Vista manages applications differently from other platforms; so, the CA SSO Client also behaves differently when you try to launch applications.

On Vista, applications are classified as standard applications and elevated applications. *Standard* applications are applications that do not need to modify or write to system files. *Elevated* applications are applications that may need to modify or write to restricted system files. As a standard user on Vista, you can run standard applications only. To run elevated applications on Vista, you must have administrator privileges.

Whenever you try to access an elevated application, Vista displays a User Access Control dialog prompting you for consent to run applications in elevated mode. You may be prompted with a UAC dialog twice based on your application resource settings set in the CA SSO Policy Manager.

To launch applications on Vista

1. Log into CA SSO.

   The CA SSO LaunchBar or CA SSO Tools displays a list of applications you can access.

2. Click an application from the application list.

   Based on the settings configured for your application, the CA SSO client does one of the following actions:

   - Displays a User Access Control (UAC) dialog prompting you for consent to run the selected application in an elevated mode.
   - Launches the selected application or a login dialog if you are accessing the application for the first time.

3. (Optional) Enter the administrator credentials in the UAC dialog, and click OK.

   Note: This step is required only if a UAC prompt is displayed on step 3.

   Based on the settings configured for your application, the CA SSO client does one of the following actions:

   - Displays a UAC dialog prompting you to consent to launch the elevated application.
   - Launches the selected application or a login dialog if you are accessing the application for the first time.

4. (Optional) Enter the administrator credentials in the UAC dialog, and click OK.

   Note: This step is required only if a UAC prompt is displayed on step 3.

5. Enter the user credentials for the selected application, and click OK.

   The CA SSO client launches the selected application.

Chapter 2: Performing a Basic Example Implementation

Before you do anything with CA SSO, we suggest you do a simple test installation to see how the basic functionality works. We recommend that you do this in a test environment using two computers: one to install the CA SSO Server and the Policy Manager on, the other to install the CA SSO Client.

To demonstrate CA SSO functionality, this chapter guides you through installing its primary components on Windows, adding a test application and then launching the test application as if you were an end user. We recommend that you work through this chapter in the order it is written.

This section contains the following topics:

Install the CA SSO Server

To demonstrate how CA SSO works, this topic explains how to install the CA SSO Server. For the purposes of this test installation, we suggest you use a test computer with a Windows operating system.

To install the CA SSO Server

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select CA SSO Server.

3. Click Install and select the Typical installation.

   Note: During the installation process, you will create an CA SSO Server Administrator and an LDAP Directory Administrator. The details for the CA SSO Server Administrator will be used when you create a test user.

More information:

Create a Test User (see page 49)

Install the Policy Manager

To demonstrate how CA SSO works, this topic explains how to install the Policy Manager, which is a management interface for administrators. For the purposes of this test installation, install this on the same system with the test CA SSO Server.

To install the Policy Manager on a test computer

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Configuration Tools, Policy Manager.
3. Click Install and accept the default installation.

Install the CA SSO Client

To demonstrate how CA SSO works, this topic explains how to install the CA SSO Client. The CA SSO Client is installed on each user's computer. For the purposes of this test installation, do not install this on the same computer as the CA SSO Server.

Note: To install CA SSO Client on Windows Vista, elevation of privileges is required.

To install the CA SSO Client on a test computer

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Client.
3. Click Install and follow the prompts.

4. When prompted to create a server set, enter the following details:

   Server Set name

   Test installation

   CA SSO Server

   The name of the test computer where the CA SSO Server is installed

   Authentication Method

   SSO

   Note: We do not recommend that you use the SSO authentication method in a production environment if you intend to use Microsoft Active Directory as your user data store.

Create a Test User

To demonstrate how CA SSO works, this topic explains how to create a test user called Bill Webb.

To create a user

1. Launch the Policy Manager and connect to the CA SSO Server.

   Note: Use the CA SSO Server Administrator username (by default this is ps-admin) and password details created while installing the CA SSO Server.

2. Select the Users icon in the left pane.
3. Expand the User Datastores and select ps-ldap
4. Right-click in the right pane and choose New, User.

   The Create New User – General window appears.

5. Enter the following details:

   User name

   Bill

   Last name

   Webb

6. Click Browse to select an authentication method.

   The Select Authentication Method(s) window appears.

7. Make SSO a selected method then click OK.
8. Click Change Password.

   You must choose an authentication method you are setting a password for.
9. Select SSO and enter and confirm a password.

   You must remember this password when you log on to CA SSO.
10. Click OK twice to confirm your new user.

    You can see the new user in the list.

More information:

Install the CA SSO Server (see page 47)

Create a Test Application

To demonstrate how CA SSO works, this topic explains how to create a test application. This will be a password-protected Microsoft Office Word document. In a real installation, this will be any application you choose to add to CA SSO.

To create a password-protected Microsoft Word document

1. Open a Word document and type "This document is password protected".
2. From the Tools menu, select Options and then select the Security tab.
3. Enter the password "Secret1" in the Password to Open field and click OK.

   The Confirm Password window opens.
4. Confirm your password and click OK.

   Make sure you remember this password.
5. Save your document as "SSO Test".

   For the purposes of this exercise, we assume the document is stored in the C:\ drive. You can change this location but you must change the example script used later in this chapter.
6. Re-open the document to test that it requires a password.

Create a Test Script

To demonstrate how CA SSO works, this topic explains how to create an SSO logon script to launch the test application. This topic uses the example application that you created in the previous topic.

To create a simple SSO script

1. Launch the application you want to write a script for.
2. Make a note of the Window that prompts you for input.

   Here are the details for our test application:

   Location and name

   C:\SSO Test.doc

   Title of the prompt window

   "Password"

   Buttons on the Window

   OK, Cancel

   Input required by the user

   Type password and press OK

3. Create the SSO logon script.

   For more detailed information, see the *Tcl Scripting Reference Guide*.

   Here is one for our test application that you can use. You may have to change the location:

   ```
   sso run -path "C:\\Program Files\\Microsoft Office\\Office11\\Winword.exe" -args {"C:\\SSO Test.doc"}
   sso window -title "Password"
   sso type -text "$_PASSWORD"
   sso type -text "{enter}"
   ```

4. Save the test script as a plain text file in the scripts directory on the CA SSO Server computer and name it. For this example, SSO Test Script, and save it to the following directory:

   ```
   \Program Files\CA\Single Sign-On\Server\Scripts
   ```

Add the Test Script to the Policy Manager

To demonstrate how CA SSO works, this topic explains how to define the test script that you just created, on the CA SSO Server, using the Policy Manager and assign that application to the test user, Bill.

To define a test script on the CA SSO Server

1. Launch the Policy Manager.
2. Select the Resources icon in the left pane.
3. Navigate to Single Sign-On Resources, Application Resources, Application.
4. Right-click in the Application Window and choose New.

   The Create New APPL Resource – General dialog appears.
5. Fill in the details of the application.

   For example:

Name:	SSO Test Script
Caption:	Password Protected Word Doc
Type:	Desktop Application

   Note: The caption is what the user sees in their CA SSO Application List.
6. Click Scripting.

   The Scripting dialog appears.
7. Enter the name of the script you created previously in the Script File field, and then click OK.

   For example, SSO Test Script.
8. Select the Authorize icon.

   The Create New APPL Resource – Authorize dialog appears.
9. Right-click in the Add/Edit/Delete Access Rules window and choose Add Rule.

   The Add Rule dialog appears.
10. Choose your test user, Bill. When you are finished click OK twice to return to the main window.

    You have now created added your SSO script and assigned it to your test user.

Log onto the SSO Client and Launch the Test Application

To demonstrate how CA SSO works, this topic explains how to log onto the SSO Client and launch the test application, which simulates the end user experience. After you launch the application, you can explore the SSO Client using the Status Icon from the tray menu, the Launchbar, and SSO Tools. This will help you decide which parts of SSO functionality you want to enable, and which parts you want to restrict.

To log onto the SSO Client and launch the test application

1. Launch the SSO Client from the Windows Start menu

   Start, All Programs, CA, Single Sign-On, CA Single Sign-On Launchbar

   The SSO Launchbar appears.

2. Click Logon and use the following details:

   Server Set

   Test installation

   Auth. Method

   SSO

   User Name

   Bill

   Password

   SSO Password that you created for this user

   You should see the test application.

   You may need to run "Refresh Application List" from the Options menu on the SSO Client interface.

3. Click on the test application.

   The Set Login Information box appears.

4. Enter the Login name and password for the test document and click OK.

   CA SSO logs you into the password protected Word document called "SSO Test" and stores that password on the SSO Server. This means that the user is never prompted to enter their password for this application again. For our example, we used the password "Secret1".

5. Close the Word document, then re-launch the test application from the Launchbar.

6. This time you are logged into the open document but are not asked for a password.

Chapter 3: Project Management

This section contains the following topics:

Establish Implementation and Business Teams

As with any other implementation project, the success of the CA SSO installation at your site depends on human factors: the skills and performance of the implementation team and the cooperation of the end users.

Before any serious deployment of new technology can begin, you must assemble the proper implementation teams to facilitate the roll out of CA SSO within the business. Although you may have the actual vendor or a contractor run the project for your company, you should always understand the implementation and have an internal team assigned to work with the deployment vendor.

We recommend that you have two implementation teams, one for the technical deployment of CA SSO, and the other for the roll out within the business.

This section describes the ideal members of the technical implementation team and the business implementation team. These will vary between companies and are designed to be general guidelines. Sometimes one person can perform more than one role.

Members of the Technical Implementation Team

All implementation team members should review the CA SSO documentation set. They should also refresh their knowledge of the relevant aspects of the site’s hardware and software. The implementation team should include the following team members:

A Project Manager

Owns the overall project management tasks, deliverables, communications, and timetables.

An Architect

Owns the planning and design phase of the implementation. This team member is responsible for designing the server farm structure and well as the SSO Client and authentication configuration. They make sure that the CA SSO system can integrate with any existing software or hardware within the organization. An architect is also involved in planning the implementation roll out.

A Security Administrator

Owns the review and approval of design documents, architecture, and naming standards as they pertain to user IDs and resources. This team member is also responsible for the formation and distribution of audit reports. After the implementation is complete, the security administrator is responsible for the enforcement of the security policies and procedures established for CA SSO.

A Password Administrator

Owns and sets password security. This role might be combined with the security administrator.

An Application Administrator

Owns the end user applications within the company. This team member will understand and document the logon process for each application and work closely with the script developer who will need this information to create the logon scripts.

A Script Developer

Owns the script development and creates the Tcl scripts that let end-users log on to applications. The staff responsible for writing logon scripts for CA SSO should become familiar with *Tcl Scripting Reference Guide* and should begin writing practice scripts as soon as possible.

Technical Support Representative

Owns technical issues that arise from the installation. Staff who install CA SSO need to be familiar with migration considerations and with the steps required to install CA SSO. Administrators who maintain the SSO databases must be familiar with CA Access Control and CA Directory.

A CA SSO administrator

Owns the day to day administration of CA SSO. This person will change user passwords, assist users with problems, add and remove users from the system and may set password policies and manage user sessions.

Members of the Business Team

For best results the business implementation team should include the following team members.

All team members should be given a demonstration of CA SSO and should be familiar with the basic benefits of installing CA SSO. Stakeholders should also be reassured, where necessary, about the minimal impact on end-users. Members of this team should be encouraged to read this guide.

Management

Responsible for involvement and approval of senior management at every step of the way. This team member should be in a high enough position in the organizational structure to have jurisdiction over all the parties involved in the deployment of this technology.

A security implementation forces cooperation between corporate areas that may never have been forced to work together before. This cooperation, critical to the successful implementation of a security product, provides another reason why you need a clearly defined management commitment to the security implementation.

Operations and Technical Support Representative

Responsible for the day-to-day operation of CA SSO in terms of the hardware, software, and procedures required to maintain the service levels agreed on. The Operations group is also responsible for disaster recovery, business continuum, failover, and backups.

Network and Systems Representative

Responsible for maintaining the connectivity of the environment in which CA SSO runs. Since there are several components of CA SSO that can reside in multiple systems across the network, it is important to include these groups in the design and architecture phase of the implementation. During this implementation phase of CA SSO, you need to consider firewalls, protocols, DMZ, operating systems, authentication server, servers, and so on.

Business Representative

Responsible for the policies that affect the end user’s experience with certain business applications.

End User Liaison

Represents the end users experience when it comes to interface decisions and user awareness issues. This person should have full voting rights when deciding what the user sees and what procedures get implemented that will directly affect the experience of an end user.

Trainer

Works with the technical project manager and the end user liaison to develop and deliver training to end users.

Establish Project Objectives

This section describes how to establish project objectives.

Define Project Objectives

The project objectives should include the following:

- Define CA SSO security objectives and select CA SSO functionality
- Define CA SSO performance and scalability objectives
- Map and document the computing environment, including users, data stores and applications
- Prepare the implementation plan, which includes defining the CA SSO databases and user data stores
- Prepare the performance and scalability testing in the test environment to ratify your optimal server configuration
- Install and configure servers
- Define security rules including primary authentication and application authentication
- Populate the CA SSO databases
- Prepare and test Client install packages
- Create and test the logon scripts
- Test the implementation
- Train end users to use the CA SSO Client

Formulate a Security Policy

CA SSO provides a solution for security and productivity problems that result from users having to work with many different passwords. Like any security solution, CA SSO is most effective when it is integrated into a well-defined and comprehensive system security plan.

CA SSO implementation should conform to system security requirements regarding overall system security policies, password policies (either present policies or new, stronger policies that can take advantage of CA SSO features), physical protection of servers and backup servers, and auditing. In addition, general system requirements regarding response time and survivability should be considered when planning the number, location, and general configuration of CA SSO Servers and backup servers.

The initial assignment of the security implementation project team may be to develop and recommend the security policy or the document of security objectives for your environment. You may be able to use or borrow concepts from the established policies within your company with the same generic security requirements, such as authentication and authorization.

If the security policy or the document of security objectives has already been developed, the implementation team can use this document as its mandate. If these documents must be developed, the team is an ideal committee to do it since they can take into account the concerns of each affected area while developing the objectives. If each area agrees to the direction being set, which is more likely with active participation, then implementation can proceed smoothly without time-consuming discord among the business areas.

After the security policy has been formulated, upper management should issue a position statement to all internal employees and appoint a security officer (or at least a security administrator). The security officer can then ensure that employees are made aware of the security policies and procedures that they must adhere to and the consequences of any security violation.

Plan the Implementation

You should always install and test a new system in a controlled environment. Here are the suggested steps involved with the CA SSO implementation.

- Plan the implementation
- Implement a Test bed installation
- Conduct a Pilot Test

- Prepare the installation environment
- Deploy CA SSO
- Conduct End User training

Phases of the Implementation Plan

Although CA SSO installation is straightforward and flexible, it is affected by, and affects, much of your site’s system. You need an implementation plan to schedule and control the properly paced introduction of CA SSO into the nodes of the network and into the procedures of the workplace. For efficiency, the plan has to provide step-by-step procedures, guidelines, and timetables.

The Initial Planning Session

An initial planning session should define the CA SSO configuration. All the relevant servers and clients should be identified, together with the users and the applications to be secured. Relationships between applications and users have to be mapped.

Once decisions have been made on configuration, the team has to detail each of the stages of implementation.

The plan should also take into consideration any other significant events, such as installation of new hardware or software, that is planned for the same period and which could affect implementation.

We recommend that you define a pilot group that will have CA SSO installed first. A pilot group can provide valuable initial experience that can prevent problems in the full-scale implementation. You should make a decision about the size and location of the pilot group and the applications to include in the pilot study.

Once the implementation plan is finalized, the team should prepare a project schedule for the pilot and final implementation.

In a large computer system, it is not practical to implement CA SSO for all applications and for all users in one stage. An advantage of CA SSO is that it allows for phased implementation, staggered by groups of users and/or groups of applications. The implementation team has to set priorities for adding user groups and application groups.

Project Management

Implementing CA SSO is a major project. As with any major endeavor, you need to follow good project management guidelines to ensure a successful implementation.

In addition to creating an implementation team, you need to:

- Hold regular meetings
- Establish an archive of all pertinent documentation relating to the implementation
- Review your corporation’s security policies and procedures

Collect Data

Before a detailed plan can be formulated, the implementation team will have to collect considerable relevant information. The team has to map and document the computing environment, in particular those elements that directly affect CA SSO implementation.

It is essential that the data about system configuration, operating systems, applications, and authentication methods be detailed and up to date.

Use a form or checklist to collect information in a systematic way.

Here is a list of the information that you need to obtain. The scope and detail of initial database planning depends on the scope of the final implementation project itself. It is important to define the entities shown in the following table.

Entity	Definitions must include
All the applications to be accessible using CA SSO	■ Application name/identifier ■ Application path ■ Application host ■ Authentication method ■ The application group to which the application belongs, if any
All the authentication hosts that will be used by CA SSO	■ Authentication method ■ Authentication host names ■ The authentication host group to which the authentication host belongs, if any
All the authentication host groups (if authentication host groups are planned)	■ Authentication host group name ■ Authentication host names of the

Entity	Definitions must include
	authentication hosts that are to be linked to the authentication host group
User groups planned	■ User group name ■ The names of users in the group

Implement a Test Bed Installation

Before you move into the Pilot Testing Phase, install and configure the CA SSO system within a Test environment to make sure of all the components are configured correctly. This step facilitates the smooth introduction of CA SSO to users within your company and help with user-acceptance, as well as assisting the implementation from a technical perspective.

Conduct a Pilot Test

In large systems, installation of the SSO Clients on end-user computers begins with a pilot group.

When a pilot test is to be run, SSO Clients are first installed on the pilot group's computers. The implementation team will works closely with the pilot group for testing and for obtaining end user feedback. It is important to prepare testing procedures and worksheets for recording results.

Every user has to be authorized to use the specific method of authentication. Generally, we recommend that you set the user's AuthMethod token value to SSO when first implementing CA SSO. This enables you to test the validity of the records in the USER and APPL classes, without being affected by any problems in primary authentication installation.

However, once in production, the token must be set to its planned value. For example, to enable an end user to use Windows authentication, you must:

1. On the SSO Client machine, open the Auth.ini file.
2. In [ServerSet1]\AuthMethods, type "WIN".
3. In [ServerSet1]\AuthWIN, type the name(s) of the Windows authentication computer(s).

For example:

```
[ServerSet1]
Name=Logon at work
PolicyServers=Server01
Authemthods=WIN
…
AuthWIN=Server02
```

Note: For the above example to work, the WIN authentication agent must be installed and configured, and the CA SSO Server have an authhost defined for the WIN authentication agent.

Perform Authentication Test

The following procedure describes how to verify that the WIN authentication works, as in our example.

Perform post-installation verification

1. Click Start, Programs, CA, Single Sign-On , CA Single Sign-On Launchbar.
2. Click Logon.
3. Select WIN in the Authentication Method in the drop-down selection field.
4. Click Log On.
5. Enter username and password then click OK.

   If you successfully authenticate to the CA SSO Server, then the WIN authentication method is verified.

Prepare the Installation Area

Before you begin the CA SSO installation, you should review and prepare the intended site. This stage, which can also be referred to as a walk-through, involves the implementation team arriving on site to review the equipment and facilities for the subsequent stages. Successful completion of this stage should be viewed as a prerequisite to continuing the implementation.

The site staff should provide information about the hardware and software on the site. The implementation team should check technical details of servers, end-user workstations, and primary authentication systems against the preliminary data already received and analyzed.

The team should look for potential obstacles and problems. Hardware and software prerequisites should be checked, including:

- All client workstations must be correctly configured to use the network
- Each CA SSO component (clients, servers, authentication hosts) should be able to ping its peer by name
- If you are using Windows authentication, CA SSO users should have a domain account and logon rights
- Any third-party authentication software to be used (for example, RSA SecurID), should be properly installed and configured
- All machines requiring software install must have either a DVD drive, or be able to copy the installation files from a network location.

Deploy CA SSO

In the production phase, the CA SSO Client software is installed on all the end-user workstations group by group (either by geographical groups or by business function groupings). If there is no pilot testing phase, it may be advisable to check the work of the previous stages by installing the SSO Client on one or two workstations in each user group.

During each phase, auditing data and user feedback are collected and analyzed. This allows management to evaluate the success of the implementation and indicates what adjustments have to be made.

During this stage, the implementation team will begin transferring responsibility for routine administration of CA SSO to the site’s IT organization.

Conduct End User Training

CA SSO implementation requires only minimal end-user training. Before implementation, end users should be told that changes in the network will automate their logging into password-protected applications. They need to be informed on how the specific implementation on the site affects them in regard to system logon, first-time CA SSO logon, routine logon to applications, logon to sensitive applications, station lock release, re-authentication, and password change.

End users should also be informed where they will still be asked for passwords (such as for sensitive applications and password changes), they need only their user ID, a primary authentication password, and, where applicable, an additional biometrics or token authentication. In addition, end users should be informed that when they log onto applications for the first time using CA SSO, they might be required to provide their application password to the SSO Server.

Following installation of SSO Clients, end users must be told where to find the CA SSO application list and the various ways of starting applications.

If CA SSO is implemented together with new third-party authentication, new password rules and/or other security policies, then end users must be educated on these topics.

Chapter 4: Designing the CA SSO Architecture

This chapter explains how to design your CA SSO architecture. You should design your architecture and set it up in a test environment before implementation.

You must design your CA SSO architecture so that you can determine how many servers you will need, how they need to be configured and where they need to be located.

Because hardware and operating systems change constantly, it is not possible to recommend a specific configuration. Instead, this chapter gives you information to help you design the right architecture for your organization and walks you through the architecture for a test company and explains why they designed their architecture in a particular way.

This section contains the following topics:

Pre-Design Considerations

Before you implement CA SSO you need to first determine some factors that affect how you set up your architecture.

Environmental Constraints

This section explains what environmental information you need before you begin designing your CA SSO architecture.

- How many workstations will the CA SSO Client be installed on?
- How many business sites do you have?
- How geographically dispersed are the users and servers?

- How many CA SSO Servers will be installed? This is often a balance between the competing requirements for redundancy, performance and cost.
- What kind of authentication do you currently use and what authentication servers will CA SSO communicate with?
- Where do you currently store user data?
- Do you currently have a hardware load balancer (HLB)?
- Do you currently have a firewall or intelligent DNS installed?
- What applications will be added to CA SSO, and will therefore require SSO Scripts? This includes the complexity of the scripts needed.
- Do you currently have an older version of CA SSO installed?

Performance Requirements

Before you design your architecture, you need to define your performance requirements. You need to consider:

- Continuity of data so that if any server fails, no data is lost
- Performance continuity so that if a computer fails users can still access their applications
- Acceptable response times for users logging on at peak times
- Acceptable response times in the event of a total data center failure
- Synchronization and dissemination of new data, such as when new applications are added to the system
- Whether it is acceptable for users to enter their application passwords once when CA SSO if first implemented, or once each time you upgrade CA SSO from a previous version

Geographic Locations

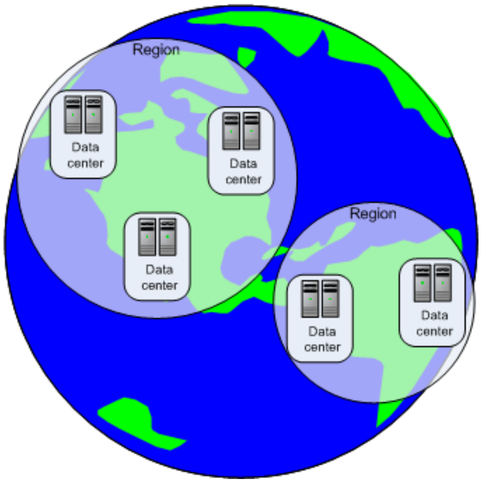
The size and geographic distribution of your company affects how you should architect CA SSO. This chapter uses the following terminology when referring to geographic distribution.

Data center

A data center typically is an office or server room that houses IT infrastructure. When we refer to a data center, we expect that you will have two or more SSO Servers in every data center in a server farm configuration.

Region

A region is a cluster of data centers. A region is typically defined by country or group of countries. For example, all data centers in Europe might be considered one region, and all data centers across the Americas might be considered another region.



Server Farms

Each data center has an SSO Server farm. This consists of at least two servers configured to share data. Servers within a server farm are called peers.

When you have multiple server farms and want to configure replication between those server farms, you should assign one server in each server farm to be the "hub". The hub server is the server that receives incoming updates from external server farms, and propagates the data to its peers within its own server farm.

Data that Needs to be Replicated

This section explains what CA SSO information needs to be stored and how that information should be propagated within server farms (within data centers) and between server farms (within a region).

Token Information (PSTD DSA)

When a user logs on, the SSO Server creates a token which stores information about the user’s session and stores this in the token data store. The token data store is often referred to as the PSTD and the data changes frequently. Typically a token expires after a day. Tokens are specific to a user on a specific workstation.

To prevent a single point of failure and eliminate the need for the user to re-authenticate during the day if a server fails, the token data store should be replicated to all servers in the server farm.

This replication means that any CA SSO Client can get logon information from any server in the server farm.

You would not replicate the token data store between server farms (within a region) because it would create unnecessary network traffic for minimal gain. The only benefit of this replication would be that if an entire server farm failed, users would not have to reauthenticate when their SSO requests were serviced by another server farm (within the region) or by another region.

Logon Information (PS DSA)

Each user has their own logon information: the usernames, passwords and other data that is used to log them into their SSO-enabled applications. This is referred to as LoginInfo. This logon information is relatively static and is specific to an individual user.

To let users log onto CA SSO quickly in any office within a data center, you should replicate user logon information between servers within a data center. This might affect a user who regularly worked from different offices within the same city.

You should replicate logon information between data centers so that users who travel to different regions can access their logon information and so that the system has failover if an entire data center fails. You might choose to replicate this information less frequently, such as once a week during an off peak time, to save on network traffic, because you would not expect users to travel between regions on a daily basis.

SSO Scripts

Each SSO-enabled application has an SSO script written for it. SSO Script information is relatively static. It would only change when a new application was added to the company or to the CA SSO system.

Your own application distribution and the rate of change dictate how frequently this information is replicated to each server farm, in your CA SSO architecture. You may choose to use the Windows Task Scheduler to regularly copy all script files from a central management server.

PSBGC

PSBGC information is information about application lists that is stored in a background cache on the SSO Server. PSBGC information is relatively static. It would only change when a new application was added to a user's application list.

You may choose to use the Windows Task Scheduler to regularly run the PSBGC service on all SSO Servers, or, alternatively, on one SSO Server and then copy the PSBGC cache information to all other SSO Servers.

CA Access Control Rules

The CA Access Control data store contains all authorization rules and authentication method information.

The CA Access Control's Policy Model Data Base (PMDB) feature will replicate this data between the members of the server farm. The replication between server farms could be set up via “subscribing” the “hubs” to one another.

Load Balancing

Load balancing is a networking concept that allows network traffic to be distributed amongst servers, thus balancing the load. Load balancing can take place in software or in a physical device (hardware load balancing).

The CA SSO architecture is designed to take advantage of such technologies. This section contains descriptions of networking concepts used in describing the CA SSO architecture.

Hardware Load Balancer

A Hardware Load Balancer is a physical device that directs clients to individual servers in a network, based on factors such as server processor utilization, the number of current connections to a server or the overall server performance. The use of a hardware load balancer minimizes the probability that any server will be overwhelmed and optimizes the network bandwidth available to each computer.

Intelligent DNS

Intelligent DNS is an advanced DNS feature which allows the DNS to route client requests based on geographic proximity. For example, a request initiating from an CA SSO Client in California (US) is first routed to the geographically closest CA SSO Server: within California if one is available, then an CA SSO Server in New York, then one outside the US.

CA SSO Client Failover

The CA SSO Client has in-built failover, between the CA SSO Client and the authentication host, and between the CA SSO Client and CA SSO Server.

When connecting to an authentication host for authentication, the CA SSO Client is able to failover between authentication hosts. The CA SSO Client tries the first authentication host listed for that authentication method in the [ServerSet] section of the Auth.ini file; if this authentication host does not respond within a specified time, the CA SSO Client contacts the next one in the list, and so on.

When connecting to an CA SSO Server, the CA SSO Client attempts to contact the first CA SSO Server in the list. Just like with authentication hosts, the Client tries the second CA SSO Server if the first does not respond within a specified time, it tried the next one.

Load Balancing for CA SSO

CA SSO supports three forms of fault tolerance and load balancing:

- Intelligent DNS (iDNS) and Hardware Load Balancer (HLB)
- Hardware Load Balancer (HLB) only
- Client-based failover

We recommend the first option, especially in a large-scale deployment because it provides the best end user experience.

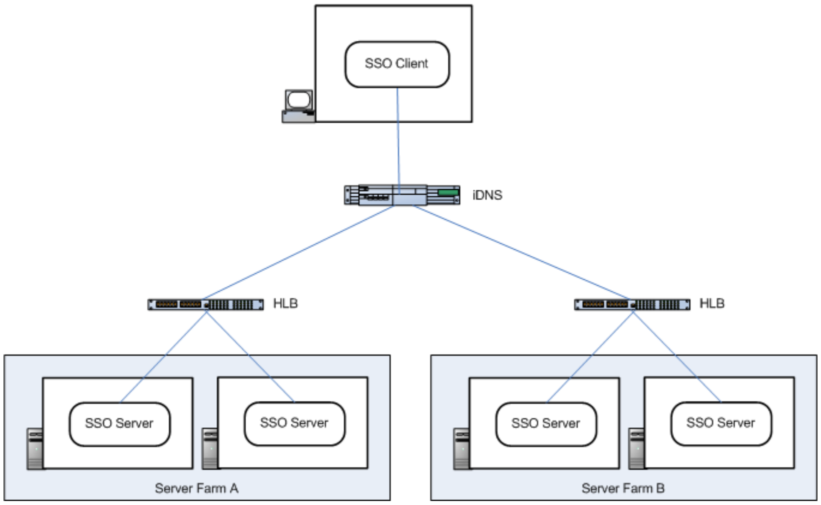
Intelligent DNS and Hardware Load Balancer

In this configuration the CA SSO Client has a single CA SSO Server DNS name defined: the iDNS server. The iDNS system directs the CA SSO Client to the closest server farm based on the CA SSO Client's IP address. Each server farm has a HLB in front of it which directs the connection to the appropriate CA SSO Server within the server farm (based on availability and load).

The iDNS manages failover between server farms and the HLB manages failover and load balancing within server farms.

The advantages of this configuration are:

- All CA SSO Clients in the network are configured the same way
- The HLBs take care of failover and load balancing between servers in a server farm
- The iDNS routes each CA SSO Client to the nearest server farm and provides failover between server farms (in the event that a data center goes offline).



If you plan to use the WIN authentication method, you need to define an alias on the authentication agent host computer in order for the CA SSO Client to recognize the return address, which it expects to be the HLB machine.   This is because the WIN authentication method uses named-pipes for communication, a requirement of which is that the receiving server has the name the CA SSO Client expects. If this goes through a HLB, the CA SSO Client request will have the HLB as the authhost name.

To do this, add the following registry key:

Path

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\lanmanserver \parameters

Key name

OptionalNames

Key type

REG_MULTI_SZ

Key value

load balancer name, *iDNS name*

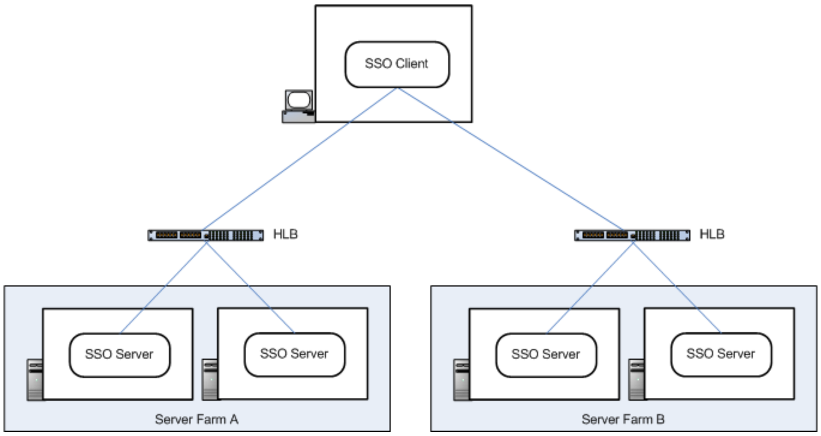
Hardware Load Balancer Only

In this configuration the CA SSO Client is configured with the DNS name for the HLB in front of each server farm. The CA SSO Client connects to the HLB and then the HLB directs the connection to the appropriate server (based on availability and load).

The advantage of this configuration is that the HLB takes care of failover and load balancing between servers within a server farm, and the Client's failover functionality takes care of failover between server farms.

The disadvantage of this configuration is that a specific Auth.ini file must be created with the name of each server farm (HLB) in priority order.

For example, 50% of Clients may have Server Farm A listed first, then Server Farm B. The remaining CA SSO Clients would have Server Farm B listed first, then Server Farm A.



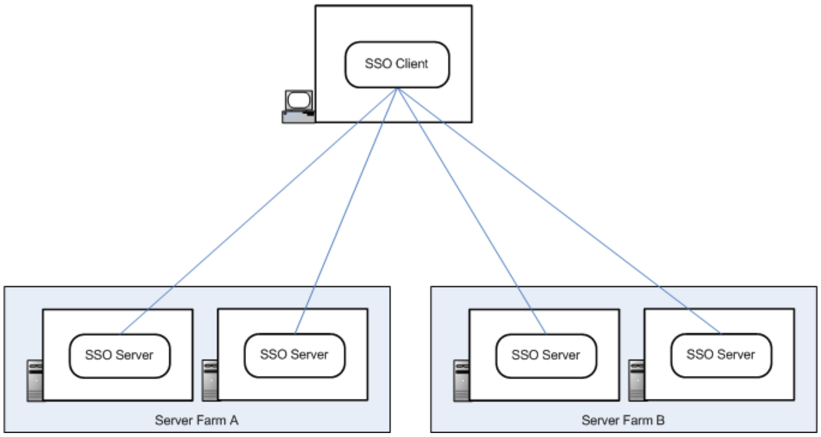
CA SSO Client-Based Failover Only

In this configuration the CA SSO Clients are configured with the name of each server in failover order.

The advantage of this configuration is that no additional hardware is required.

The disadvantages of this configuration are that:

- Dynamic load balancing within a server farm is not available
- A specific Auth.ini file must be created for each server farm, optionally with the name of backup server farms, in priority order
- The user experience may be affected by delays in the event of server failure



Sample Architecture

The number of servers that you install in each server farm depends on the number of users you have in that location and what your performance, fault tolerance, and failover requirements are.

There is no rigid rule to determine the numbers of servers you need, but we strongly recommend that you always use a minimum of two servers in a server farm configuration at every location to provide backup if one server ever fails.

The following section in this chapter looks at the server configuration choices made by a sample company called ABCcorp which may give you some insight.

ABCcorp Architecture

ABCcorp has the following data centers in their European region:

- Paris (5,000 users)
- Rome (18,000 users)
- Dublin (12,000 users)

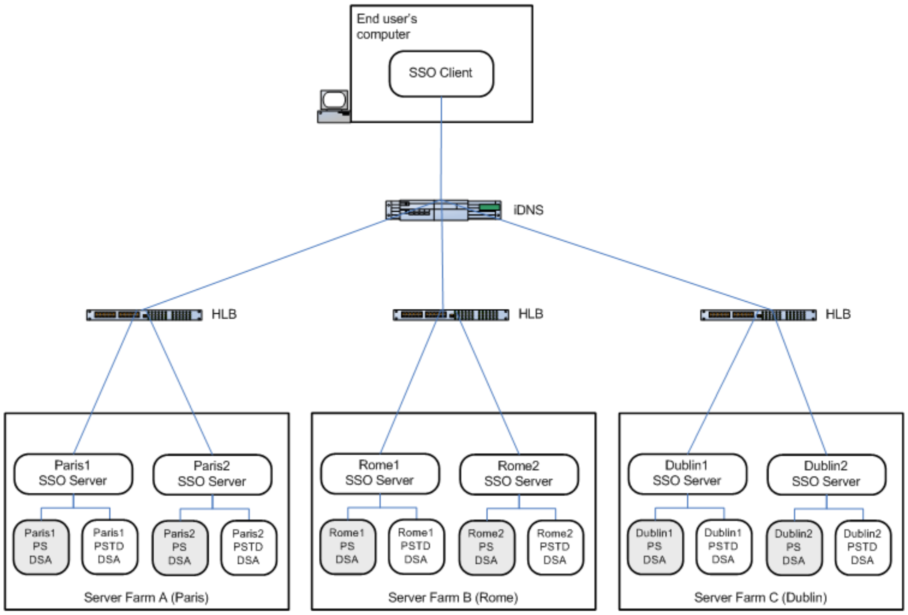
ABCcorp has the following additional technology:

- intelligent DNS (iDNS)
- hardware load balancer (HLB)

Basic ABCcorp architecture consists of three server-farms. Each of the data centers has one server farm. Each server farm has two CA SSO Servers. ABCcorp applied a formula of 10,000 users per server as a starting point, and then verified whether their performance and reliability requirements were being met when they did scalability testing. Even though their Paris office only has 5,000 users, they still need two servers in that data center to provide failover in the event that one server fails.

The CA SSO Clients in ABCcorp connect to the CA SSO Servers via an iDNS which helps route requests to the closest server farm, and a HLB which then directs each request to the CA SSO Server that is least busy.

The following diagram shows the CA SSO architecture for ABCcorp. This diagram only shows logon information data (PS DSA) and token information (PSTD DSA).



How Logon Information Data is Replicated

ABCcorp want to replicate the logon information stored in the Paris server farm. They want to replicate this information to the other server within the server farm, as well as replicating the information to other server farms. This provides failover: if an entire server farm fails users can continue to use CA SSO uninterrupted because their logon information is available from the other server farms in the region.

The following process explains how logon information that originates on one of the Paris servers is replicated to its peers within the server farm, as well as to all the other servers in the region in Rome and Dublin.

To implement this behavior, the logon information data stores (PS directory on each CA SSO Server) must be configured to perform Multiwrite Group replication (**multi-write-group**). Multiwrite replication uses Directory System Protocol (DSP) to chain updates between servers in real time.

Multiwrite Group replication works when DSAs are configured into **multiwrite groups**. When an update is applied to a DSA belonging to a group, it passes the update to other DSAs in the group. Multiwrite Group replication uses DSP chaining to apply updates to all replication peers in real time. When an CA SSO Client makes an update request, that update is applied immediately to the local DSA, and then all other DSAs and Group DSA's.

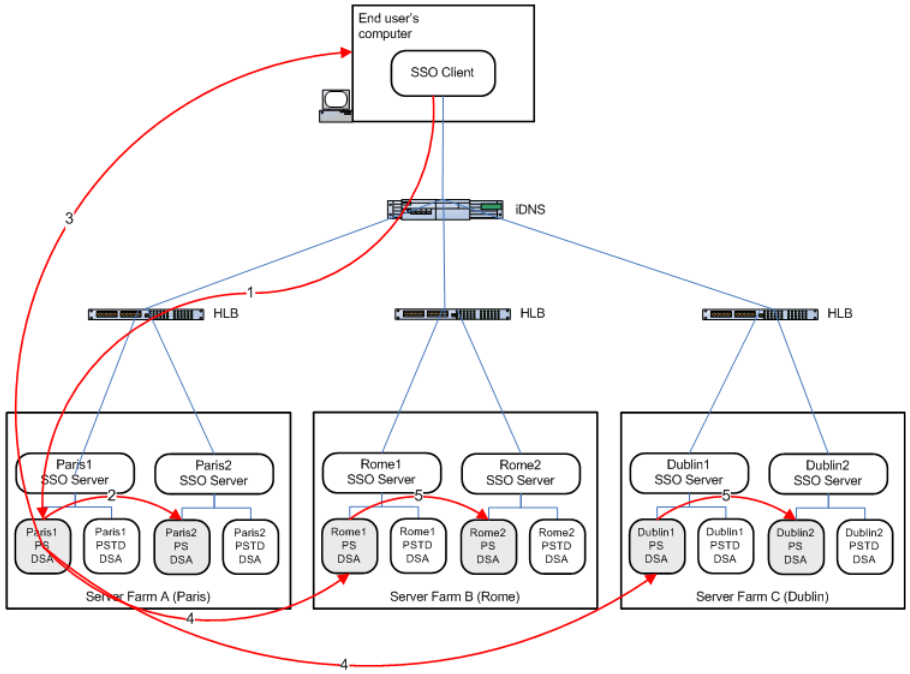
For more information about Multiwrite replication or Multiwrite Group replication, see the *CA Directory Administration Guide.*

In ABCcorp’s system, the CA SSO Client only receives a response from the CA SSO Server when the Server’s DSA receives a confirmation response that all other DSAs in the group have successfully applied the update.

The following process explains how this works for ABCcorp.

1. The CA SSO Client sends a request that the iDNS routes to Paris1 Server. As the server processes the request it updates its login information data store: Paris1 PS DSA. When the DSA receives this request it applies this update to itself.
2. The Paris1 PS DSA then copies this data to its peers within the server farm (Paris2 PS DSA). This should be done using multiwrite. If these updates succeed the peers send confirmation to Paris1 PS DSA.
3. Paris1 PS DSA then sends confirmation to the CA SSO Server which then responds to the CA SSO Client.

4. Paris1 PS DSA then sends the update to the hub DSAs in the other server farms in the region (Rome1 PS DSA and Dublin1 PS DSA).
5. The hub DSAs then send the information to their peers within their server farm. Each hub DSA sends the request to the other PS DSA in its group. When each hub DSA (Rome1 PS DSA and Dublin1 PS DSA) has received confirmation from each peer in its **multiwrite group**, it sends the confirmation response to the first DSA (Paris1 PS DSA). The Client confirmation is not affected by possible delays in the network between server-farms, because these updates are asynchronous.



Paris1 PS DSAs, Rome1 PS DSA and Dublin PS DSA should all be set up for multimaster replication. This means that if a request is sent to Dublin1 CA SSO Server, it is propagated to all of the other DSAs in the Dublin server farm and to the Hub DSAs of other server farms.

For more information on Multimaster replication, see the *CA Directory Administration Guide.*

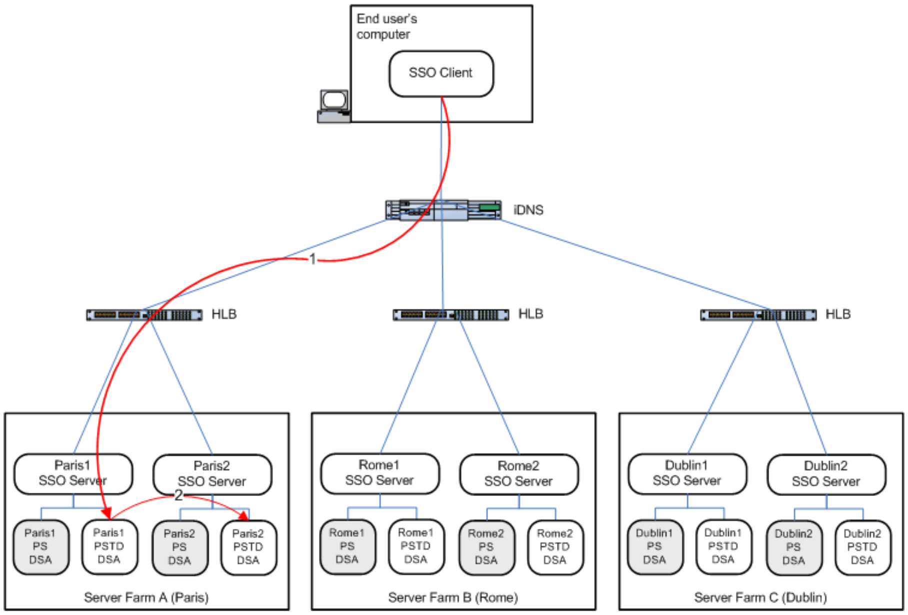
How Token Directory Information is Replicated

ABCcorp wants to replicate the token directory information within each server farm but does not want to replicate this information between server farms. This is because the information in the token directory is related to the CA SSO Client sessions; if this is not replicated, an CA SSO Client request going to the second server within a farm will need to generate a new token with the Server.

The following process explains how token information that originates on one of the Paris servers is replicated to its peers within the server farm, but not between different server farms.

For more information about **multi-write-group** configuration and updates queues, see the *CA Directory Administration Guide* chapter on Replication.

1. The CA SSO Client sends a request that goes to Paris1 PSTD DSA which applies this update to itself.
2. Paris1 PSTD DSA copies this information to Paris2 PSTD DSA.



Password Changes

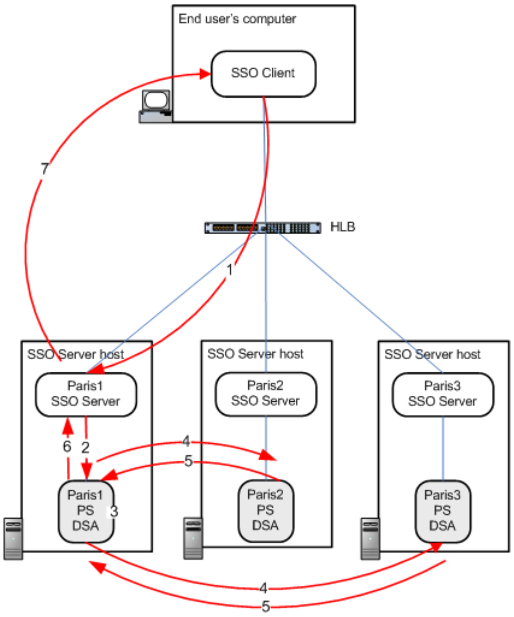
Propagating user password changes is an important function within CA SSO. This section explains how password changes are handled in various situations.

How Passwords are Changed With All Servers Running

This section describes how an application password change is made with all CA SSO Servers in a server farm running (normal conditions).

1. The CA SSO Client sends a password change request, which the HLB directs to Paris1.
2. Paris1 checks the password against the password policy, and if valid tells Paris1 DSA to change this user's password.

3. Paris1 DSA writes the change locally, verifying that it is a valid change.
4. Paris1 DSA communicates this password change request to its peers in the server farm. (Paris2 DSA and Paris3 DSA).
5. Each DSA makes the change and returns successfully.
6. Paris1 DSA returns success to Paris1 CA SSO Server.
7. Paris1 CA SSO Server returns success to the Client.

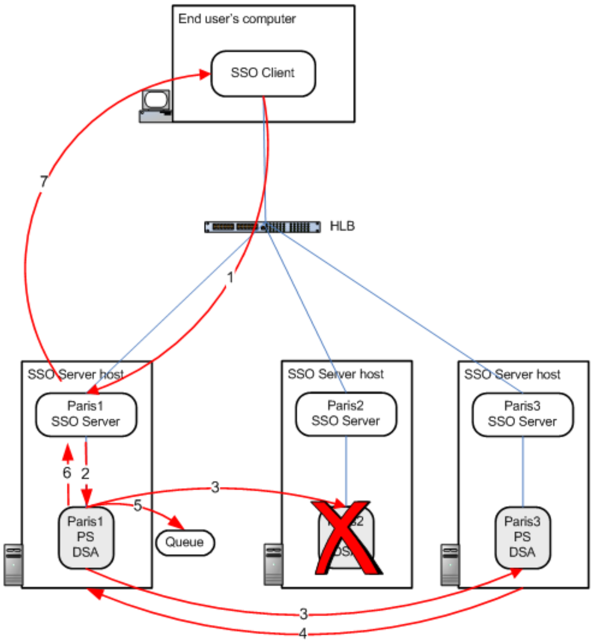


How Passwords are Changed With the Second DSA Down

This section describes how an application password change is made with the second DSA in the server farm not running.

1. CA SSO Client sends a password change request, which the HLB directs to Paris1.
2. Paris1 checks the password against the password policy, and if valid tells Paris1 DSA to change this user's password.

3. Paris1 DSA writes the change locally, verifying that it is a valid change then sends the change to its peers (Paris2 DSA and Paris3 DSA).
4. Paris3 DSA makes the changes and returns success. Paris2 does not respond.
5. Paris1 DSA queues the change request to Paris2 DSA.
6. Paris1 DSA returns success to Paris1 CA SSO Server.
7. Paris1 CA SSO Server returns success to the Client.

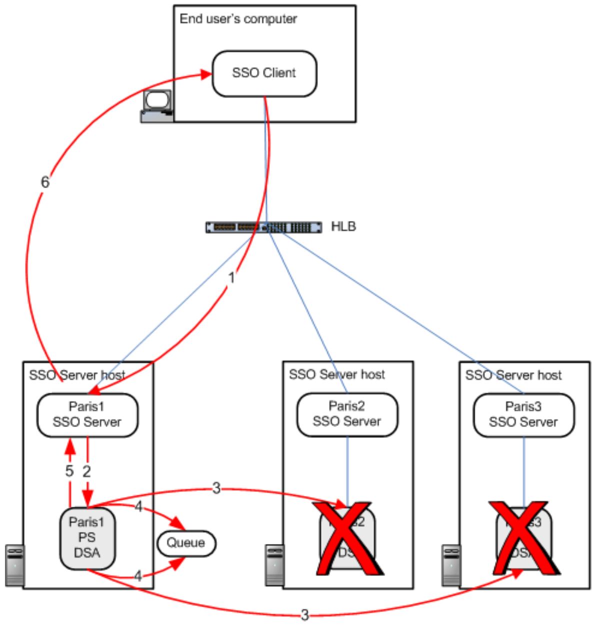


How Passwords are Changed with Second and Third DSAs Down

This section describes how an application password change is made with the second and third DSAs in the server farm not running.

1. CA SSO Client sends a password change request, which the HLB directs to Paris1.
2. Paris1 checks the password against the password policy, and if valid tells Paris1 DSA to change this user's password.
3. Paris1 DSA writes the change locally, verifying that it is a valid change then sends the change to its peers (Paris2 DSA and Paris3 DSA).
4. Neither Paris2 DSA nor Paris3 DSA respond, so Paris1 DSA queues the change request to Paris2 DSA and Paris3 DSA.

5. Paris1 DSA returns success to Paris1 CA SSO Server.
6. Paris1 CA SSO Server returns success to the Client

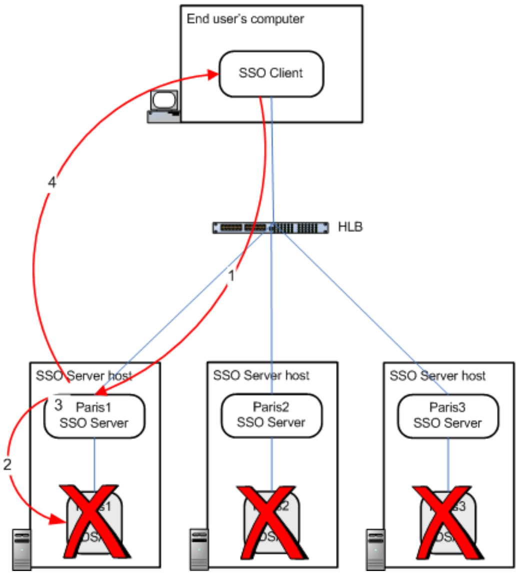


How Passwords are Changed When All DSAs are Down

This section describes how an application password change is made with all DSAs in the server farm not running.

1. CA SSO Client sends a password change request, which the HLB directs to Paris1.
2. Paris1 checks the password against the password policy, and if valid tells Paris1 DSA to change this user's password.

3. None of the DSAs respond.
4. Paris1 CA SSO Server returns failure to the Client.

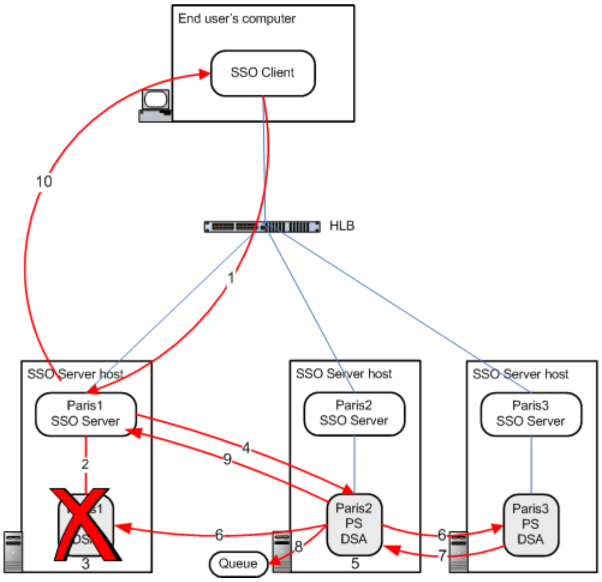


How Passwords are Changed when the Local DSA is Down

This section describes how an application password change is made with the local server's DSA not running.

1. CA SSO Client sends a password change request, which the HLB directs to Paris1.
2. Paris1 tries to check the password against the password policy, and if valid tell Paris1 DSA to change this user's password.
3. Paris1 DSA does not respond.
4. Paris1 then sends the change request to one of the other DSAs in the farm, in this case Paris2 (this can be configured to send to a DSA in a different server farm).
5. Paris2 writes the change locally, verifying that it is a valid change.
6. Paris2 then sends the change to its peers (Paris1 DSA and Paris3 DSA).

7. Paris3 DSA makes the changes and returns success.
8. Paris1 does not respond so Paris2 DSA queues the change request to Paris1 DSA.
9. Paris2 DSA returns success to Paris1 CA SSO Server.
10. Paris1 CA SSO Server returns success to the Client.



Performance Tuning

After you design your CA SSO architecture, implement this design in a test environment. In your test environment, we recommend that you simulate an expected user load.

As part of this process you may need to tune certain settings to achieve optimal performance of your CA SSO system.

There are a number of settings that you need to understand and determine so that you can optimize your CA SSO installation. This section is a summary of what you need to understand and how you should determine those settings.

The rest of this document explains each of these settings in more detail and guides you through how to determine the optimal value for each setting.

Connection processing

An understanding of how the CA SSO Server processes connections is a critical component of determining the parameters which impact performance of the CA SSO Server.

Fork limit

The server's optimum fork limit value for a particular system should be determined empirically by simulating a load and observing the throughput and response times for various values of fork limit.

Server idle timeout

The CA SSO Server idle timeout value can be determined from the results of the fork limit test according to the following formula:

Server idle timeout = (fork limit / operations per minute) * 60

PSTD size

The CA SSO Server's maximum cache size should be determined by simulating a load and observing the memory usage of the PSTD on the CA SSO Server hosts.

Receive queue size

The CA SSO Server's receive queue size can be calculated to optimize the minimum-connection-delay.

Client response timeout

The CA SSO Client's response timeout is a subjective value. It should be set to a value that reflects the system response time expectations.

Connection Processing

To understand the performance constraints, you must understand the way in which the CA SSO Server handles connections as well as the system parameters which contribute to the performance and scalability behavior.

How Connections are Processed

When an CA SSO Client connects to the CA SSO Server, the actions and subsequent response of the server depend on the number of existing connections to the server. The CA SSO Server handles incoming connections as outlined below:

1. The first connections are accepted and processed immediately. The number of connections is set using the fork limit setting. Each of these connections is allocated to a thread for servicing their requests.

2. Once the fork limit has been reached, subsequent connection requests are added to the receive queue, until it becomes full.
3. Once the receive queue limit has been reached, subsequent connection requests are added to the busy queue. In this case, CA SSO Clients will immediately receive a "Server Busy" response. As server busy response is sent immediately the busy queue does not usually reach capacity.
4. If the busy queue does reach capacity, the CA SSO Server is unable to respond to the connection requests. In this case, the operating system directs a number of connections to the TCP/IP queue.
5. If the TCP/IP queue limit has been reached, connection requests will receive a TCP/IP error.

One Thread Per Connection

CA SSO operates on a one-thread-per-connection model: when an CA SSO Client makes a connection to the CA SSO Server, a thread is allocated to that connection. Once an CA SSO Client has a connection, it can make any number of requests using that connection. The connection is held open until either the CA SSO Client or the CA SSO Server closes the connection.

The CA SSO Client explicitly closes the connection after it has finished its request.   In previous versions of CA SSO, the CA SSO Client did not close the connection: the CA SSO Server closed the connection only after the Server Idle Timeout has been reached.

Connection Rate

The rate at which CA SSO Clients make connections to the server is the connection rate. The connection rate has two aspects:

- the sustained connection rate – the sustained rate of connection requests that the server can respond to
- the instantaneous connection rate – the number of connection requests that occur at any one time

The sustained connection rate is the average number of simultaneous connections the server can handle over an extended time period. The fork limit determines the sustained throughput rate. The underlying system hardware and resources determine the overall system capability. The fork limit should be set to make best use of the system resources. If the fork limit is too small, the system is underutilized. If the fork limit is too large, the system thrashes.

We recommend that you test your environment to determine your best fork limit. If you do not set the size of your fork limit to the best value for your system, your system will not be fully utilized.

The instantaneous connection rate is the connection rate at any instant in time. If many connection requests are made, not all of them can be handled instantly. Additional requests may be queued until the server can respond to them, as described in How Connections Are Processed.

The size of the receive queue affects the system’s response to the instantaneous connection rate. Depending on the throughput rate, the receive queue size should be large enough to handle the largest instantaneous connection rate (number of simultaneous connections).

However, the user experiences queued connection requests as delays in CA SSO Client operations. For example, if the server can handle 1000 connection requests per minute and the receive queue contains 5000 connection requests, the connection requests at the end of the queue may experience a delay of several minutes before they are handled. The CA SSO Client will not wait for five minutes, instead it times out a request after two minutes if no response is received.

If this delay is considered unacceptable, the client response timeout in the CA SSO Client can be set so that the request is cancelled more quickly and the user is shown an appropriate message.

System Parameters

To determine the optimum configuration for system performance, you must understand certain parameters available on the CA SSO Server and the CA SSO Client.

Fork limit

The fork limit parameter determines the number of concurrent threads the CA SSO Server has to immediately process connections.

The greater the fork limit, the more concurrent connections can be handled. However there will be a limit imposed by system resources because the machine can run out of resources if there are too many threads in use. For example, the machine may spend all of its time swapping between threads, or there may be so many threads that little work is done on each thread per unit of time.

Change the Fork Limit

As part of tuning your CA SSO system for optimal performance you might want to change the Fork Limit.

To change the Fork Limit

1. Launch the Policy Manager
2. From the left pane, select Resources
3. Navigate to Configuration Resources, Policy Server Settings, Communication

   The View or Set GPSConfigProperty Properties - Settings window appears.
4. Find ForkLimit setting and adjust the number.
5. Click OK twice to save changes.

Note: This is deliberately set to a low value out-of-the-box. This is so that if the CA SSO Server is running on a machine with very little memory, the Server will not cause the system to thrash. You must increase the fork limit on your server to the limit found during your testing. This will depend on various factors including hardware your CA SSO Server is running on, your network speed, and your system architecture.

Receive Queue Size

The size of this queue is defined by the receive queue size parameter in the CA SSO Server. If no value is defined on the CA SSO Server for the receive queue size, the default is the fork limit multiplied by 10. Requests in the receive queue will be handled by the server as they are removed from the queue. Depending on the throughput, this will be experienced as a delay in the CA SSO Client.

If the receive queue size is large, connections may be queued on the server waiting to be handled, which will result in the user experiencing a poor server response. For example, if the server is handling 1000 connection requests per minute and the receive queue size is 5000, those connection requests at the end of the queue may experience a delay of several minutes before they are handled.

If the receive queue size is small, some users will receive a “CA SSO Server unavailable” response if their request is further back in the queue. A better outcome may be achieved by increasing the receive queue size, which would result in more connection requests being successfully handled without error messages. The cost is that there may be a longer delay for any response for some of the users, depending on where their requests are in the queue.

Change the Receive Queue Size

As part of tuning your CA SSO system for optimal performance you might want to change the Receive Queue Size.

To change the Receive Queue Size

1. Launch the Policy Manager
2. From the left pane, select Resources
3. Navigate to Configuration Resources, Policy Server Settings, General

   The View or Set GPSConfigProperty Properties - Settings window appears.
4. Find ReceiveQueueSize setting and adjust the number.
5. Click OK twice to save changes.

TCP/IP Queue Size

The TCP/IP queue size represents the number of connection requests that the operating system will queue while the CA SSO Server is unable to respond to requests. The size of this queue can be set in the Policy Manager by the CommListenQueueSize setting.

Change the TCP/IP Queue Size

As part of tuning your CA SSO system for optimal performance you might want to change the TCP/IP Queue Size.

To change the TCP/IP Queue Size

1. Launch the Policy Manager
2. From the left pane, select Resources
3. Navigate to Configuration Resources, Policy Server Settings, General

   The View or Set GPSConfigProperty Properties - Settings window appears.
4. Find CommListenQueueSize setting and adjust the number.
5. Click OK twice to save changes.

Client Response Timeout

Client response timeout is a configuration that can be set on the CA SSO Client. If the CA SSO Server does not respond to a request before this timeout, the CA SSO Client displays an error message. This may happen if the request is in the CA SSO Server queue but the load is such that the CA SSO Server cannot handle the request quickly enough for the CA SSO Client.

Change the CA SSO Client Response Timeout

As part of tuning your CA SSO system for optimal performance you might want to change the CA SSO Client Response Timeout.

To change the CA SSO Client Response Timeout

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [NetworkCommunication] Section.

3. Edit the following value:

   ConnectTimeout

   Defines the time in seconds the Client will try to connect to the SSO Server before it gives up.

   Value: *time in [days]d[hours]h[minutes]m[seconds]s*

   Default: 120s

4. Save the Client.ini file.

Server Idle Timeout

Server idle timeout is a configuration setting on the server. The CA SSO Server holds open its connection with the CA SSO Client until after the last operation until this timeout is reached. This is so that subsequent CA SSO Client requests will be processed quickly. A reconnection cost is incurred if another request is made by that CA SSO Client after the connection is closed. See One Thread Per Connection.

Change the Server Idle Timeout

As part of tuning your CA SSO system for optimal performance you might want to change the Server Idle Timeout.

To change the Server Idle Timeout

1. Launch the Policy Manager

2. From the left pane, select Resources

3. Navigate to Configuration Resources, Policy Server Settings, Communication

   The View or Set GPSConfigProperty Properties - Settings window appears.

4. Find TimeOutRecv setting and adjust the number.

5. Click OK twice to save changes.

Chapter 5: Implementing the CA SSO Server

This section contains the following topics:

About the CA SSO Server

The CA SSO Server is the center of CA SSO. It resides on a Windows server and manages resources and provides services to the CA SSO Client.

The CA SSO Server performs the following functions:

- Authorization:
  - Builds the list of applications that a user is allowed to access and sends it to the CA SSO Client
  - Controls who can access the data held in the CA SSO data stores
  - Controls when data can be accessed

- Policy and user management:
  - Manages users and resources in:
    - CA Access Control
  - Manages users in
    - CA Directory
    - Active Directory and other third party data stores
  - Provides the logon scripts and the user-specific logon data for each application
  - Sets and clears passwords
  - Configures Session Management and offline application support
- Auditing, logging and tracing

About CA SSO Server Farms

A server farm is a system of multiple networked CA SSO Server computers. If you need more than one CA SSO Server within your company you should connect them together in a server farm. The data on each server can then be replicated to all servers in the farm.

The benefits of a server farm that has full replication and hot backup include:

- No need to maintain separate data stores
- Failover, which is the ability of a server to take over if one server goes offline, without affecting services

More information:

Load Balancing (see page 71)

Implement a Server Farm

The purpose of a server farm is to enable each server to send data to, and receive data from, every other server in the farm to allow backup and failover.

If you are installing a new server farm and you have no existing CA SSO Servers, you can install all the CA SSO Servers from the installation CD and specify each of the other servers in the server farm to automatically set up a server farm. After all the CA SSO Servers have been installed in this way, they will automatically communicate with each other and replicate data.

Note: To set up a server farm, ensure you select the install option of *Custom* in the CA SSO Server installation wizard.

Before You Install

The Before You Install section is designed to guide you through what you need to know before you install the CA SSO Server. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Decide on Method of Installation

This section explains each type of installation to help you choose which method you should use.

The CA SSO Server can be installed using:

Graphical installation wizard

The installation wizard leads you through the various steps required for installing the CA SSO Server. Use this method to familiarize yourself with the installation options.

Silent installation

Using the command line, you can silently install the CA SSO Server.

If you choose to do a silent install, you must specify the variables by either:

- Creating a response file
- Using command line parameters

Note: The Windows command line may limit the number of characters in a single command. For complex or detailed command line installations, you should consider using a response file.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the CA SSO Server:

- Ensure that all system requirements are met before you begin installing the CA SSO Server. For a complete list of system requirements, see the *SSO Readme* file.
- Ensure you are logged in as an administrator before installing the CA SSO Server.
- Ensure you know all relevant information prior to running the installation, including:
  - Administrator details for the CA SSO Server and LDAP user stores.

    Note: A default user name is provided during the wizard install.
  - If you are implementing the CA SSO Server on a server farm, you need to:
    - Provide the name of each server
    - Ensure all servers are connected to the network and available to each other at install time
    - Ensure that a proper IP naming resolution system (preferably DNS configured as primary resolver) is in place and is capable of forward and backward hostname/FQDN to IP Address lookup of all server farm members.

    Note: To install the CA SSO Server on a server farm, you need to select the *Custom* install option.
- If you intend to install the CA SSO Server and the Policy Manager on the same computer, see Policy Manager and CA SSO Server on One Computer for more information.
- If a complex password policy is set on a Windows computer, the CA SSO Server installer may automatically generate a password for its ps-pers account that fails to meet the requirements of the password policy. In this case, the setup wizard displays an additional page which lets the administrator manually enter a password for the ps-pers account.

  Note: This password is used only when the automatically generated password fails. Ensure you contact your system administrator for information about minimum password length and password complexity requirements. For silent installs you can use the *-W ps-per.password* command.

- When installing CA SSO Servers in a server farm environment make sure you use the same password for the ps-pers accounts on all computers within the server farm.
- Ensure that your operating system produces a reliable and correct timestamp for the local time-zone.   If it does not, the product may not work. For example, the operating system clock of an CA SSO Server host in New York is set to US Eastern Daylight Time (EDT), while the operating system clock of an LDAP Authentication Agent host in San Francisco is set to US Pacific Daylight Time (PDT).

More information:

Policy Manager and CA SSO Server on One Computer (see page 118)

Install the CA SSO Server

This section tells you how to install the CA SSO Server Windows platforms.

Install the CA SSO Server using the Wizard

This topic explains how to install the CA SSO Server on a server farm using the Product Explorer.

To install the CA SSO Server using the wizard

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Server.
3. Click Install and follow the prompts.
4. Accept the default install option of Custom. Alternatively, if you want to install a standalone version of the CA SSO Server, select Typical.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install the CA SSO Server Silently

You can install the CA SSO Server silently. This means that you need to provide the information that would normally be supplied by the administrator during the graphical wizard installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the install wizard. You can find the command line setting required for accepting the license agreement and silently installing the CA SSO Server at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256-character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer Wizard automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select CA SSO Server.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the CA SSO Server using silent installation.

4. Open a command prompt and navigate to the CA SSO Server folder on the product DVD.

5. From the command prompt, type:

   ```
   setup.exe -silent -V LICENSE_VIEWED=value {parameters}
   ```

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

setup Command—Install CA SSO Server

The command line parameters for installing the CA SSO Server include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-W license.selection

Specify whether you accept the license agreement displayed in the SSO Server wizard install wizard.

The command has the following format:

```
-W license.selection={Value}
```

Value: 0 | 1 | 2

- 0 - Nothing is is selected
- 1 - Terms of the license agreement are accepted
- 2 - Terms of the license agreement are not accepted

-W setupTypes.selectedSetupTypeId

Specifies whether you want to proceed with a typical or custom install. Select custom if you want to install the SSO Server on a server farm.

The command has the following format:

```
-W setupTypes.selectedSetupTypeId={Value}
```

Value: typical | custom

-W destination.policyServerDestination

Defines the install location for the SSO Server.

The command has the following format:

```
-W destination.policyServerDestination=[Value]
```

Value: The path to the SSO Server.

-W destination.accessControlDestination

Defines the install location for CA Access Control.

The command has the following format:

-W destination.accessControlDestination={*Value*}

Value: The path to CA Access Control.

-W destination.directoryDestination

Defines the install location for CA Directory.

The command has the following format:

-W destination.directoryDestination={*Value*}

Value: The path to CA Directory.

-W destination.ingresDestination

Defines the install location for Ingres.

The command has the following format:

-W destination.ingresDestination={*Value*}

Value: The path to Ingres.

-W destination.ingresDBDestination

Defines the install location for the Ingres database.

The command has the following format:

-W destination.ingresDBDestination={*Value*}

Value: The path to the Ingres database.

-W ps-admin.username

Specifies the user name of the SSO Server administrator. This information is used in SSO Policy Manager to administer the SSO Server.

The command has the following format:

-W ps-admin.username={*Value*}

Value: User name of the SSO Server administrator.

-W ps-admin.password

Specifies the password of the SSO Server administrator. This information is used in SSO Policy Manager to administer the SSO Server.

The command has the following format:

```
-W ps-admin.password={Value}
```

Value: Password of the SSO Server administrator.

-W ldap-admin.username

Specifies the user name of the LDAP directory administrator. This information is used to access the LDAP directory.

The command has the following format:

```
-W ldap-admin.username={Value}
```

Value: User name of the LDAP directory administrator.

-W ldap-admin.password

Specifies the password of the LDAP directory administrator. This information is used to access the LDAP directory.

The command has the following format:

```
-W ldap-admin.password={Value}
```

Value: Password of the LDAP directory administrator.

-W ps-pers.username

Specifies the user name of the SSO Server personality user.

The command has the following format:

```
-W ps-pers.username={Value}
```

Value: The user name of the SSO Server personality user.

-W ps-pers.password

Specifies the password of the SSO Server personality user.

The command has the following format:

```
-W ps-pers.password={Value}
```

Value: The password of the SSO Server personality user.

Note: When installing the SSO Server, the installer automatically generates a password for the user. However, the password generated may fail on machines with strong password complexity requirements. The *ps-pers.password* command lets you provide a password that meets your password policy requirements. This password is used only when the automatically generated password fails. Ensure you contact your system administrator for information about minimum password length and password complexity requirements.

-W server-farm-members.hostlist

Specifies the hostnames of the servers making up the server farm.

The command has the following format:

```
-W server-farm-members.hostlist=[Value]
```

Value: A comma separated list of one or more names of servers (other than the present server) making up the server farm.

-W comm-mode.CommMode

Specifies the communication mode of the CA SSO Server.

The command has the following format:

```
-W comm-mode.CommMode=[Value]
```

Value: 0 | 1 | 2

- 0 - Non-FIPS mode
- 1 - FIPS-only mode
- 2 - Mixed mode

Note: In FIPS-only and Mixed modes, a trusted certificate is needed.

-W user-cert.certFilePath

Specifies the trusted certificate for communication. The file must be in Privacy Enhanced Mail (.pem) format.

```
-W user-cert.certFilePath=[FilePath]
```

FilePath: Absolute path (including the file name) of the Trusted Certificate.

-W user-cert.keyFilePath

Specifies the private key for the certificate. The file must be in Privacy Enhanced Mail (.pem) format.

```
-W user-cert.keyFilePath=[FilePath]
```

FilePath: Absolute path (including the file name) of the private key for the Trusted Certificate.

Install the CA SSO Server Silently using a Response File

Use the following procedures to install the CA SSO Server silently using a response file.

Note: You can use a combination of response file and command line options to silently install the CA SSO Server. In this case, the command line options will override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the CA SSO Server folder on the product DVD.
3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create a Response File (see page 102)
setup Command—Install CA SSO Server (see page 97)

Create a Response File

Response files record user specific install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the CA SSO Server folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record {file name}
   ```

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   A response file is created in the directory specified.

Post-Installation Configuration Options

The topics that follow describe post-installation tasks.

Add a New Server Farm Member (Windows)

To add a new CA SSO Server to an existing server farm

1. Install the new CA SSO Server on a networked machine. Ensure you select the Server Farm option and specify the other machines in the server farm.
2. Add new server details to the existing CA Access Control server farm.
3. Add new server details to the existing CA Directory server farm.

More information:

Install the CA SSO Server (see page 95)

Add New Server Details to the Existing CA Access Control Server Farm

The following procedure guides you through adding a new CA SSO Server member to the existing server farm members. This procedure needs to be repeated on all pre-existing servers in the server farm.

In this procedure, we refer to the following machines:

- Server1 = existing server farm member
- Server2 = existing server farm member
- Server3 = new CA SSO Server to be added to the server farm

Default CA Access Control database location is:

- Windows: "C:\Program Files\CA\Access Control\data\seosdb"

Default PMDB database location is:

- Windows: "C:\Program Files\CA\Access Control\data\PS_PMDB"

To add new server details to Server1 and Server2

1. Open a command prompt and navigate to the CA Access Control Bin directory.

   Note: Default install location is:

   - Windows: "C:\Program Files\CA\Access Control"

2. To list all policy model databases, type the command:

   ```
   sepmd –p
   ```

3. To list all policy subscribers (CA SSO Servers), type the command:

   ```
   sepmd –l PS_PMDB
   ```

4. To add Server3 to the list of subscribers, type the command:

   ```
   sepmd -s PS_PMDB Server3
   ```

5. To check that Server3 is in the list of subscribers, type the command:

   ```
   sepmd –l PS_PMDB
   ```

6. Start selang. Type the command:

   ```
   selang
   ```

7. Type the command:

   ```
   Env pmd
   ```

8. To accept incoming server farm updates from Server3, type the command:

   ```
   subspmd parentpmd(PS_PMDB@Server3)
   ```

9. To synchronize the seosdb with the PS_PMDB:

   a. Stop CA Access Control on the local host. Open a command prompt and type *secons -s*.

   b. Navigate to the CA Access Control data\seosdb directory.

   c. Copy all seosdb_ files into the data\PS_PMDB directory on the new server member.

   d. Start CA Access Control. Open a command prompt and type *seosd –start* (Windows).

10. To synchronize CA Access Control data between Server1/2 and Server3 (Windows only):

    a. Open command prompt on Server1.

    b. Stop CA Access Control. Type *secons -s*.

    c. Perform a backup with command:

    ```
    dbmgr –e –l –f   C:\ACdata.txt
    ```

    d. Start Access Control. Type *seosd -start*.

    e. Open a command prompt on Server3.

    f. Stop CA Access Control. Type *secons -s*.

    g. Transfer the content of directory created in step 10.c to Server 3 and replace the content of CA Access Control/data/seosdb with the data contained in back-up directory.

    h. Navigate to C:\Program Files\CA\ Access Control\bin and run the following command:

    ```
    selang –f C:\ACdata.txt
    ```

    i. Start CA Access Control. Type *seosd -start*.

    j. Repeat Steps 1 to 9 on Server2.

Add New Server Details to the Existing CA Directory Server Farm

The following procedure guides you through adding the new CA SSO Server member details to the existing server farm members.

In this procedure, we refer to the following machines:

- Server1 = existing server farm member
- Server2 = existing server farm member
- Server3 = new CA SSO Server to be added to the server farm

To add new server details to Server1 and Server2

1. On the new server (Server3), navigate to $DXHOME\Config\knowledge\.
2. Copy the PS_SERVER3.dxc and PSTD_SERVER3.dxc files to the same location on the Server1 and Server2 machines.
3. On the Server1 and Server2 machines, edit the PS_Servers.dxg file and add references to PS_SERVER3.dxc and PSTD_SERVER3.dxc.
4. Restart the PS and PSTD DSAs on Server1 and Server2.
5. To synchronise the new Server's CA Directory with the Server1/2 DSA:
   a. Stop all DSA's on Server1. Type *dxserver stop all*.
   b. Export the ps-ldap directory on Server1. Type *dxdumpdb PS_<ServerName> > filename1.ldif*

   Where

   ServerName

   Specifies the machine name of Server1.

   a. On Server3, load the exported LDIF data from Server1 into the ps-ldap directory:
      - Sort ldap.ldif. *Type ldifsort filename1.ldif   filename2.ldif*.
      - Load filename2.ldif into the directory on Server3. Type *Dxloaddb PS_<ServerName> filename2.ldif*

   b. Start all DSAs on Server1. Type *dxserver start all*.
   c. Repeat sub steps 1-4 for Server2.

Remove a Server Farm Member (Windows)

To remove an CA SSO Server from an existing server farm

1. Remove the server information from the CA Access Control server farm details.
2. Remove the server information from the CA Directory server farm details.

Remove Server Details From CA Access Control Server Farm

The following procedure guides you through removing an CA SSO Server from an existing server farm.

To remove a new server

1. Open a command prompt and navigate to the CA Access Control Bin directory.
2. To remove the server as a subscriber, type the command:

   ```
   sepmd –u PS_PMDB <server name>
   ```

   The server is removed as a subscriber to the CA Access Control server farm.

Remove Server Details From CA Directory Server Farm

The following procedure guides you through removing an CA SSO Server from a server farm.

In this procedure, we refer to the following machines:

- Server1 = existing server farm member
- Server2 = existing server farm member
- Server3 = existing server farm to be removed

To remove a server

1. On Server1 and using above naming standards, open PS_Servers.dxg.
2. Comment out the references to PS_server3.dxc and PSTD_server3.dxc, respectively.
3. Open PS_server1.dxc and comment out dsa-flags = multi-write.
4. Stop the Server1 DSA. Type the following command:

   ```
   dxserver stop PS_Server1
   ```

5. Start the Server1 DSA. Type the following command:

   ```
   dxserver start PS_Server1
   ```

6. Restart the CA SSO Server.
7. Repeat Steps 1-6 on the Server2 machine.

   The server is removed as a subscriber to the CA Directory server farm.

Configuring CA SSO to Use IPv6

Internet Protocol version 6 (IPv6) is an Internet Layer protocol for packet-switched internetworks. It is an extension to IPv4 which is currently the dominant Internet Protocol version. The Internet Engineering Task Force (IETF) has designated IPv6 as the successor to version 4 for general use on the Internet.

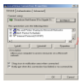
IPv6 has a much larger address space than IPv4, which provides flexibility in allocating addresses and routing traffic. The extended address length (128 bits) is intended to eliminate the need for network address translation to avoid address exhaustion, and also simplifies aspects of address assignment and renumbering, when changing Internet connectivity providers.

Installing IPv6

To install the IPv6 protocol, following this procedure:

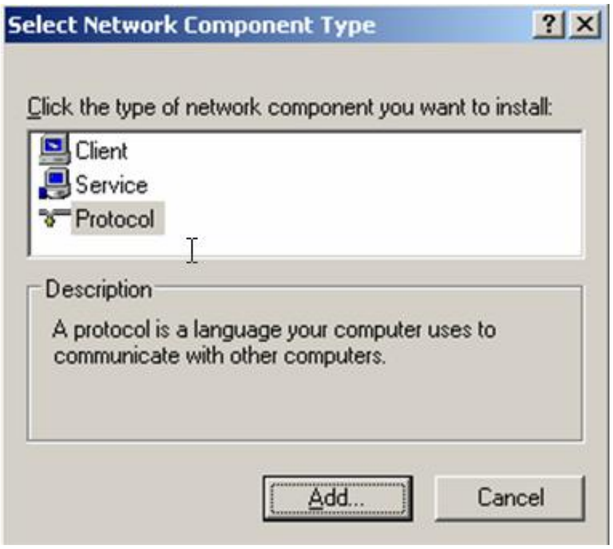
From the Loca Area Connector Properties dialog

1. Click Install



The Select Network Component Type dialog appears.

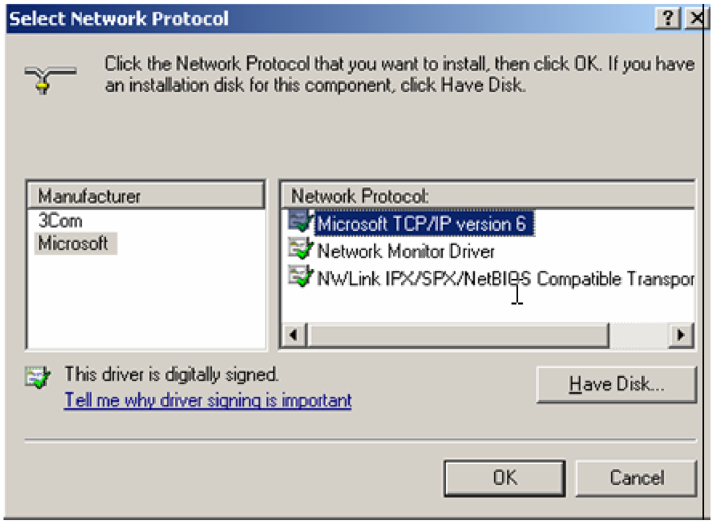
2. Click Protocol.



Select Network Component Type
?
Click the type of network component you want to install:
Client
Service
Protocol
Description
Add...
Cancel

A protocol is a language your computer uses to communicate with other computers.

The Select Network Protocol dialog appears.



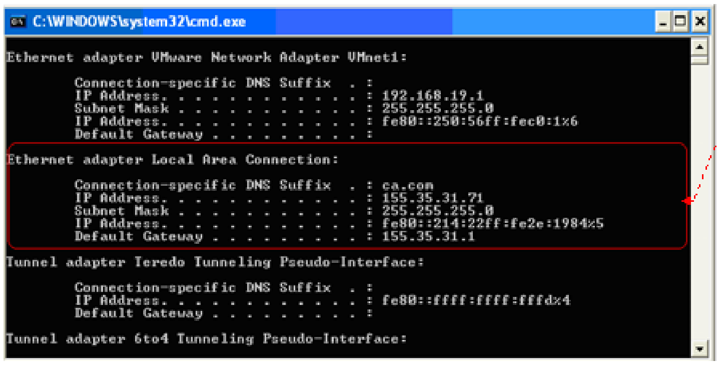
Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Select Network Protocol
Manufacturer
3Com
Microsoft
Network Protocol:
Microsoft TCP/IP version 6
Network Monitor Driver
NWLink IPX/SPX/NetBIOS Compatible Transpor
This driver is digitally signed.
Tell me why driver signing is important
Have Disk...
OK
Cancel

3. Select Microsoft TCP/IP version 6, click OK, and close all dialogs.

4. Open a console window and execute this command:

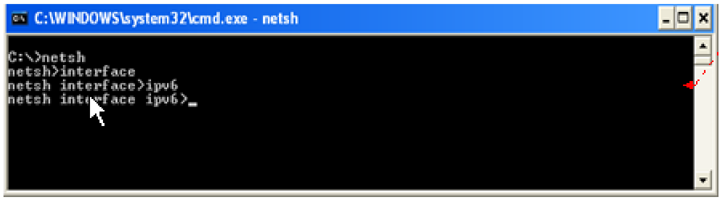
ipconfig



```
Ethernet adapter VMware Network Adapter VMnet1:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.19.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . : fe80::250:56ff:fec0:1%6
        Default Gateway . . . . . . . . . :

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : ca.com
        IP Address. . . . . . . . . . . . : 155.35.31.71
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . : fe80::214:22ff:fe2e:1984%5
        Default Gateway . . . . . . . . . : 155.35.31.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : fe80::ffff:ffff:fffd%4
        Default Gateway . . . . . . . . . :

Tunnel adapter 6to4 Tunneling Pseudo-Interface:
```

5. Enter the following command to configure IPv6:

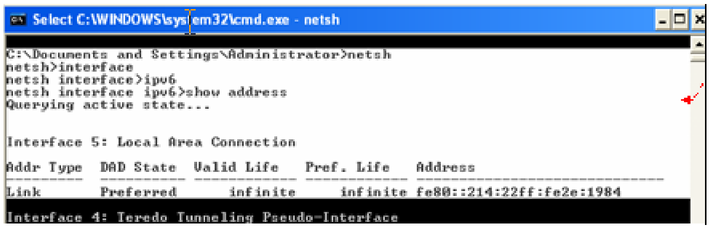
netsh



```
C:\>netsh
netsh>interface
netsh interface>ipv6
netsh interface ipv6>_
```

6. The following command shows all of the assigned v6 addresses to all of the interfaces on the system:

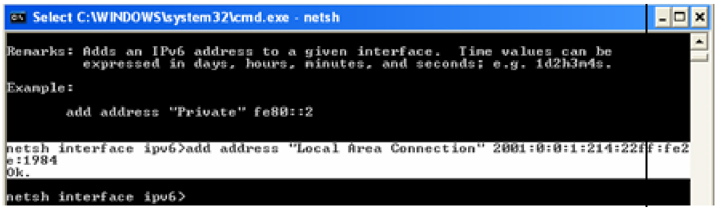
   netsh interface ipv6>show address



```
C:\Documents and Settings\Administrator>netsh
netsh>interface
netsh interface>ipv6
netsh interface ipv6>show address
Querying active state...

Interface 5: Local Area Connection

Addr Type  DAD State   Valid Life    Pref. Life   Address
---------  ----------  ------------  ------------ -----------------------------
Link       Preferred       infinite      infinite fe80::214:22ff:fe2e:1984

Interface 4: Teredo Tunneling Pseudo-Interface
```

7. The following command adds and new IPv6 address:

   netsh interface ipv6>add address



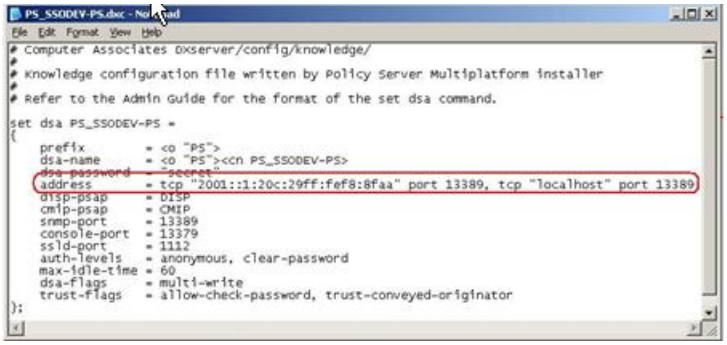
```
Remarks: Adds an IPv6 address to a given interface.  Time values can be
         expressed in days, hours, minutes, and seconds; e.g. 1d2h3m4s.

Example:

        add address "Private" fe80::2

netsh interface ipv6>add address "Local Area Connection" 2001:0:0:1:214:22ff:fe2
e:1984
Ok.

netsh interface ipv6>
```

   Note: Windows XP SP2 and Windows 2003 SP1 are not complete implementation of IPv6. Only Microsoft Vista and Windows 2008 support complete IPv6. Contact your administrator for these platforms.

Set Up CA SSO Server to Support IPv6

Once IPv6 setup is complete, install the CA SSO Server. All other components can be installed on a pure IPv6 machine as they are installed on an IPv4 machine.

A dxc file, (the configuration file for a DSA in CA Directory), has an entry “address”. The value of the address entry in the dxc specifies the directory where the IP and Port reside and where the dsa service must be listening. The PS and PSTD dxc’s need to be verified for the address value to be IPV6.



```
# Computer Associates DXserver/config/knowledge/
#
# Knowledge configuration file written by Policy Server Multiplatform installer
#
# Refer to the Admin Guide for the format of the set dsa command.

set dsa PS_SSODEV-PS =
{
    prefix          = <o "PS">
    dsa-name        = <o "PS"><cn PS_SSODEV-PS>
    dsa-password    = "secret"
    address         = tcp "2001::1:20c:29ff:fef8:8faa" port 13389, tcp "localhost" port 13389
    disp-psap       = DISP
    cmip-psap       = CMIP
    snmp-port       = 13389
    console-port    = 13379
    ssld-port       = 1112
    auth-levels     = anonymous, clear-password
    max-idle-time   = 60
    dsa-flags       = multi-write
    trust-flags     = allow-check-password, trust-conveyed-originator
};
```

Verify IPv6 Configuration

You must verify that the CA SSO Server, CA SSO Client, and SSO Policy Manager and other machines having CA SSO Software (auth agents, PSA, and so forth) are able to ping each other using IPV6 addresses. From a console window enter the following command:

```
ping <IPV6 address of remote machine>
ping –a <hostname of remote machine> (see if this resolves to IPV6 address.
```

If there is a problem, add an entry in the hosts file and recheck. Enter the following command to the respective ports to make sure communication works.

```
telnet <IPV6 address> <port number>
```

You must then verify CA SSO Server and Policy Manager configuration in IPv6 by issuing the following commands:

```
telnet <SSOServer machine IPV6 address> 13980
telnet <SSOServer machine IPV6 address> 8891
```

Note the ports being used:

- Seosagent (Access Control)– 8891
- PSTD (Token directory DSA port) – 13390
- PS(PS directory DSA port) – 13389
- SSO Policy Server (default)– 13980

Successful Telnet ensures that the Pure IPV6 communication is happening to the machines with out failure. This must ensure successful communication between the Server, Policy Manager and other components.

Note: The above verifications are related to compatible mode installation of the CA SSO Server. For verification of FIPS /DUAL modes installations on IPv6, the port numbers mentioned above must be replaced with the respective ports. Refer to "Chapter 8" of the *CA SSO Administrative Guide* for details on the port numbers.

FIPS Overview

The Federal Information Processing Standards (FIPS) 140-2 publication is a security standard for the cryptographic libraries and algorithms a product and all its components must use for encryption. Encryption affects the storage and verification of passwords, as well as the communication of all sensitive data between components of CA products and between CA products and third-party products. FIPS 140-2 specifies the requirements for using cryptographic algorithms within a security system protecting sensitive but unclassified data.

CA SSO uses the Advanced Encryption Standard (AES) adapted by the US government. CA SSO incorporates the RSA BSafe and Crypto-C ME v2.0 cryptographic libraries, which have been validated as meeting the FIPS 140-2 Security Requirements for Cryptographic Modules.

Communication

As a part of FIPS 140-2 compliance, all communication between various SSO components is encrypted using TLS/SSL protocol. The following are the SSO components that support this functionality:

- SSO Desktop Sever
- CA SSO Client
- SSO Authentication Agents (WIN, CERT, LDAP and SSO)
- Policy Manager
- PSA
- ADS Listener

AES Encryption of User and Application Passwords

To ensure FIPS 140-2 compliance, the CA SSO Server r12 encrypts all passwords (both user and application passwords) using AES 256-bit algorithm and stores them in the CA Directory database on the CA SSO Server.

To do this, the CA SSO Server maintains two encryption keys:

- The Key Encryption Key (KEK) - used to encrypt the PEK
- The Password Encryption key (PEK) - used to encrypt the user/application passwords

The installer, by default, generates these keys and places them in the server installation folder as two .key files (ssodKek.key and ssodPek.key)

However, the administrator can periodically replace these keys with newly generated ones for security reasons using the "PwdEncUtil.exe" command line utility that comes packaged with the CA SSO Server.

Note: For more information on this utility, see, "The PwdEncUtil Command Line Utility".

Also, the installer takes a backup of the initial PEK and KEK keys in the following folder to ensure that in case the keys are lost, the backed up keys can be restored and used.

```
%InstallDir%\data\Crypto
```

Note: We strongly recommend that the KEK and PEK files be backed up at a safe location all the time. The keys must also be backed up immediately whenever either of the KEK or PEK is updated using the "PwdEncUtil.exe" utility. This is very useful in restoring the old passwords when the keys are accidentally deleted or corrupted in the server installation directory.

The PwdEncUtil Command Line Utility

The command line utility, PwdEncUtil, is used by SSO administrators to:

- Generate new Key Encryption and Password encryption keys and expire/replace active keys. (The KEK and PEK that are currently being used by the CA SSO Server.)
- Re-encryption of all existing passwords after changing the PEK.
- Re-encryption of legacy passwords (After an upgrade from r8.1.)
- Verify and retain the integrity of the existing passwords if an earlier operation (such as changing the PEK or re-encrypting the legacy passwords) fails intermittently.

This utility operates in three modes:

- PEK Operations Mode
- KEK Operations Mode
- Data Check Mode

This utility is placed in %Installdir%\bin folder.

PEK Operations Mode

PEK Operations mode focuses on creation of a new password encryption key and maintenance of application passwords.

The command syntax for this mode is:

```
PwdEncUtil -p -n
```

-p

Indicates PEK operations mode.

-n

Generates a new PEK key and re-encrypts all application password encrypted with old PEK with new PEK.

KEK Operations Mode

In KEK Operations mode, a new KEK is generated and used to re-encrypt the Password Encryption Key.

The command syntax for this mode is:

```
PwsEncUtil.exe -k -n
```

-k

Indicates KEK operations mode.

-n

Generates a new KEK and decrypts the password encryption with old KEK and re-encrypts with new KEK.

Data Integrity Check Mode

In Data Integrity Check mode, the utility is used for two reasons:

- Re-encrypts all user passwords that are encrypted with the legacy SSO algorithm, with AES-256 algorithm.
- Check and retain the integrity of the existing passwords when an earlier operation (such as a PEK change or re-encryption of legacy passwords) failed intermittently or was interrupted.

The syntax for this mode is:

```
PwdEncUtil.exe -c
```

-c

Indicates data check mode.

Configuration Parameters

The SSO administrator must regularly generate new KEK and PEK keys to ensure security of the system. The following configuration parameter is stored in the Config DSA:

MaxPasswordHistoryCount

Indicates the maximum number of old passwords to be maintained in the history list. A value of 0 or a negative value is interpreted as having no limit.

Default Value: 8

Migration Considerations

The Password Encryption Utility must be run by the SSO administrator in data check mode to migrate existing passwords which are encrypted with legacy SSO algorithm to AES-256 encrypted format.

The sample command for migration is:

```
PwdEncUtil.exe -c
```

Note: You must stop the CA SSO Server services before migrating the existing passwords. When you stop the CA SSO Server services, the CA SSO Server is not available to your users, so you must plan accordingly before running the PwdEnc utility.

Chapter 6: Implementing the Policy Manager

This section contains the following topics:

About the Policy Manager

The Policy Manager is the user interface that enables you to manage the CA SSO Server and the data stores (CA Access Control and CA Directory). It is usually installed on an administrator's workstation for remote management of CA SSO Servers using TCP/IP.

You can only install the Policy Manager on Windows computers, and you can use it to manage CA SSO Servers on Windows computers. The Policy Manager can also manage multiple CA SSO Servers that are deployed in a server farm.

The Policy Manager is an administrative tool and is used by administrators to manage CA SSO information. In addition to the Policy Manager, CA SSO comes with the following administrative tools:

Session Administrator

The SSO Session Administrator is a Web administration interface. The Session Administrator lets you view and shut down users' active sessions in runtime, for example, a user calls the helpdesk and an administrator then uses the Session Administrator to close all SSO sessions that user has open. This is different from Policy Manager which lets you set session rules, for example, you want a user to only be able to have one active SSO session at a time. Once installed, the Session Administrator can be accessed from the CA folder on the Start menu.

selang

The selang command line language can be used to update the CA Access Control policy data store. However, we do not recommend using selang as your management interface. We recommend that you use the Policy Manager to configure your CA SSO environment.

pslang

The pslang command line language can be used to manage CA SSO Servers and its data stores.

Before You Install

The Before You Install section is designed to guide you through what you need to know before you install the Policy Manager. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Decide on a Method of Installation

The Policy Manager can be installed by one of three methods:

Installation wizard

The installation wizard leads you through the various steps required for installing the Policy Manager. Use this method to familiarize yourself with the installation options.

Installation wizard with custom default options

From the command line, you can pass custom default options to the installation wizard. Use this method to create a batch file that opens the wizard with the default options you want to use.

Silent installation

Using the command line, you can silently install the Policy Manager rather than just pass custom default options to the installation wizard. Use this method to push the installation to remote computers.

Policy Manager and CA SSO Server on One Computer

This section lists what you need to know if you are installing the CA SSO Server and the Policy Manager on the same computer.

Note: If you install the Policy Manager on the same computer as the CA SSO Server, you can use the Start menu shortcuts to access Start/Stop CA Access Control services and when the Policy Manager is launched it automatically connects to the local CA Access Control database.

- If you install the Policy Manager on the same computer as the CA SSO Server, you must install the CA SSO Server first.
- For silent installs, make sure that you stop the CA Access Control Engine service before you install the Policy Manager. If you have a server farm configuration, you must do this for every computer in the server farm.

Pre-Installation Checklist

Use this checklist to make sure you have performed all pre-installation tasks before you install the Policy Manager:

- Ensure that you stop CA Access Control before you install the Policy Manager.
- Ensure that all system requirements are met before you begin installing the Policy Manager. For a complete list of system requirements, see the product Readme file.
- Ensure that your CA SSO Servers have been installed.
- Ensure that you know all relevant information required for the install, including the name of the computer or computers on which you are installing the Policy Manager.
- Ensure that you have the names of the computers that host the CA SSO Servers. You need this information after the installation to connect to CA SSO Servers for the first time.
- Ensure that the computer on which you are installing the Policy Manager has TCP/IP communications with the computers that host the CA SSO Servers.

Install the Policy Manager

This section explains how to install the SSO Policy Manager.

Install the Policy Manager using the Installation Wizard

This topic explains how to install the SSO Policy Manager using the graphical installation wizard, which you can launch from the Product Explorer. You should use this method to install the SSO Policy Manager on individual computers.

To install using the installation wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Configuration Tools, Policy Manager.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of this chapter.

Install Using the Command Line

You can use the command line to:

- Pass custom default options to the graphical installation wizard
- Silently install the Policy Manager

Install Using the Command Line to Set Custom Default Options

This topic explains how to install the Policy Manager using the command line to pass custom default options to the graphical installation wizard.

Note: If you previously installed the CA SSO Server on this computer, you need to stop the CA Access Control Engine service before installing the Policy Manager.

To install using the command line

1. Open a command prompt and navigate to the Policy Manager folder on the CA SSO DVD, which is located under the config_tools directory.
2. From the command prompt, enter:

   ```
   setup.exe /s /v"[parameters]"
   ```

   /s

   Specifies whether to hide the initialization dialog.

   /v"[*parameters*]"

   Specifies the parameters to include in the silent install.

3. When the Policy Manager installation wizard opens, follow the prompts to install the Policy Manager.

More information:

Setup.exe - Install Program (see page 122)

Install Using Silent Installation

Before completing a silent install you must first read and accept the license agreement using the EULA.txt file located in the "config_tools\Policy_Manager directory on the product DVD. The command line setting required for accepting the license agreement and silently installing the Policy Manager is located at the bottom of the license agreement.

Note: If you previously installed the CA SSO Server on this computer, you need to stop the CA Access Control services before installing the Policy Manager. To do this, open the command prompt and type: *secons –s*.

To install using silent installation

1. Insert the product installation DVD.
2. Open a command prompt and navigate to the Policy Manager folder, that is "config_tools\Policy_Manager" on the product DVD.
3. From the command prompt, enter:

   ```
   setup.exe /s /v"/qn COMMAND={keyword} [parameters]"
   ```

   /s

   Specifies whether to hide the initialization dialog.

   /v

   Specifies the parameters to include in the silent install.

   /qn

   In conjunction with the /s parameter, specifies a silent installation.

   COMMAND={*keyword*}

   Defines the command required for accepting the license agreement and silently installing the Policy Manager. The actual keyword you need to use can be found at the bottom of the license agreement (EULA.txt file) which is located in the "config_tools\Policy_Manager" directory on the product DVD.

   [*parameters*]

   Specifies command line parameters to include in the install.

More information:

Setup.exe - Install Program (see page 122)

Setup.exe - Install Program

The parameters for silently installing the Policy Manager include:

/s

Specifies whether to hide the initialization dialog.

/s

/v

Specifies the parameters to include in the installation. It applies to all parameters and properties listed below except the /s command.

Place parameters within quotes ("").

/v

/L

Defines the full path and name of the installation log file. Use the mask *v to log all available information.

/L

/qn

In conjunction with the /s parameter, specifies a silent installation.

Note: You need to use the *license_accept* property to execute a silent installation.

/qn

COMMAND

Defines the command required for accepting the license agreement and silently installing the Policy Manager. The actual keyword you need to use can be found at the bottom of the license agreement (EULA.txt file) which is located in the "config_tools\Policy_Manager" directory on the product DVD.

COMMAND={*keyword*}

SSOMODE

Specifies whether the Policy Manager is used to manage CA SSO.

Only available for silent installation.

SSOMODE=[*Value*]

Value: 0|1

- 1 for yes
- 0 for no

Default: Yes

INSTALLDIR

Specifies the location where the Policy Manager will be installed.

INSTALLDIR=[*Value*]

Value: The install location.

FIPSMODE

Specifies the FIPS mode for the Policy Manager

FIPS_MODE=[value]

Value: The FIPS Mode value:

- 0 – non_fips mode
- 1 – FIPS Mode

COMM_MODE

Specifies the communication mode for the Policy Manager.

COMM_MODE=[value]

Value: The Communication Mode value:

- * non_ssl - Non-SSL Mode (in case FIPS_MODE = 0)
- * fips_only - FIPS only mode (in case FIPS_MODE = 1)

FIPSIDENTITYFILEEDIT

Specifies the path to the Certificate file (.pem file).

FIPSIDENTITYFILEEDIT=[value]

Value: The path to the Certificate file (.pem file).

Example: setup.exe

The following example sets the installation directory, and installation log file defaults for the Policy Manager installation and proceeds with the silent installation.

```
setup.exe /s /v"INSTALLDIR=C:\CA\PM FIPSMODE=1 COMM_MODE=fips_only
FIPSIDENTITYFILEEDIT=c:\cert.pem FIPSPRIVATEKEYFILEEDIT=c:\cert.key /L*v
%SystemRoot%\PMInstall.log"
```

Perform Post-Installation Verification

The following procedure describes how to verify that the Policy Manager installation is successful.

Perform post-installation verification

1. Click Start, All Programs, CA, Single Sign-On, Policy Manager.
2. Log on to the Policy Manager.
3. Click the Users, Agents, and Resources icons on the program bar. Expand the folders in the tree to verify that the basic functionality is accessible through the Policy Manager.

Chapter 7: Implementing the CA SSO Client

This section contains the following topics:

About the CA SSO Client

The CA SSO Client is the desktop component of CA SSO. You must install it on every end-user workstation. The CA SSO Client lets users:

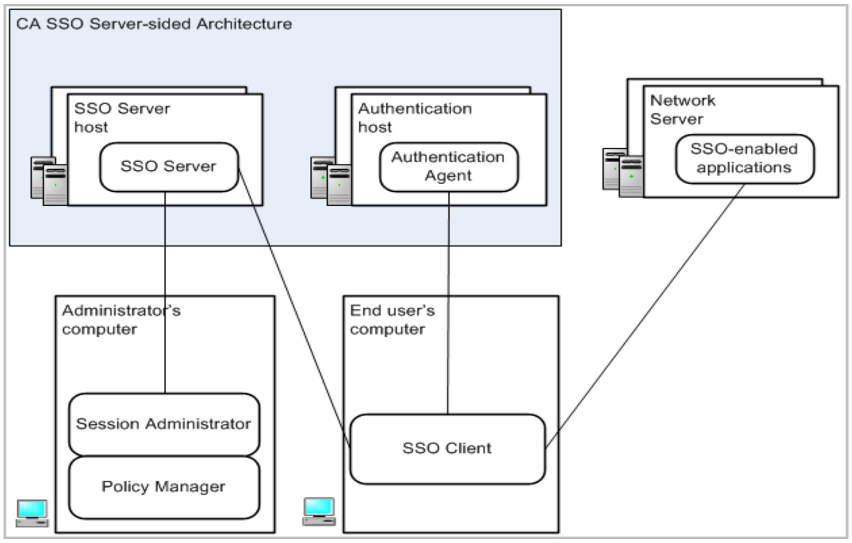
- Authenticate to Single Sign-On
- Access SSO-enabled applications
- Access SSO-enabled applications configured for offline use
- Lock their workstation (with the SSO GINA or CA SSO Credential Provider)
- Change their authentication credentials, if supported by the authentication method.

The CA SSO Client also performs actions that the user cannot see. The CA SSO Client:

- Executes SSO scripts (such as logon scripts)
- Stores authentication information
- Retrieves information from the CA SSO Server

Architecture

The following diagram shows how the CA SSO Client fits into the architecture of a typical CA SSO deployment.



CA SSO Client Installation

This section explains how to install the CA SSO Client including pre-installation considerations.

Decide Where to Install the CA SSO Client

The CA SSO Client:

- Must be installed on every end-user's computer
- Can optionally be installed on the administrators' computers

Note: You can distribute the CA SSO Client using Unicenter Software Delivery.

If you are implementing Citrix Application Migration, install the CA SSO Client on both the Citrix MetaFrame Presentation Server and the Citrix ICA Client computer. There are specific versions of the CA SSO Client for the MetaFrame Presentation Server and the ICA Client. These are different options you can select during the CA SSO Client installation.

Wizard Installation versus Silent Installation

There are two ways to install the CA SSO Client:

- Wizard installation (Windows GUI):
  This is recommended if you are installing the CA SSO Client on less than 10 computers.
- Silent installation (command line prompt):
  This is recommended if you are installing the CA SSO Client on more than 10 computers.

You need administrative user privileges to install SSO Credential Provider on a Windows Vista workstation and if UAC is enabled, a consent window is displayed before you proceed.

You should decide which installation method to use based on how many computers require the CA SSO Client. The numbers indicated above are just a guide. Each implementation has different requirements.

If you choose to do a silent install you must specify the variables by either:

- Creating a response file
- Using command line parameters

Note: The Windows command line (cmd.exe) imposes a 256 characters limit in a single command. For complex or detailed command line installations, consider using a response file.

Typical Versus Custom Installation

During the installation you are asked to choose whether you want to do a custom installation or a typical installation. You only need to select custom installation when you want to:

- Install the SSO GINA or SSO Credential Provider functionality.
- Select one of the Lock Workstation Mode options. This covers shared workstation mode and you only receive this option if you choose to install the SSO GINA or SSO Credential Provider.

- Configure the SSO Tools as the default user interface, instead of SSO Launchbar. This affects how users access their CA SSO application list.
- Create an additional shortcut for your chosen SSO interface and where this should be located.
- Install the CA SSO Client on a Citrix Metaframe Presentation server or an ICA Client computer (this is only relevant if you are deploying Citrix Application Migration with CA SSO).

Note: You need administrative user privileges to install SSO Credential Provider on a Windows Vista workstation, and if UAC is enabled, a consent window is displayed before you proceed.

Note: SSO GINA is supported only on Windows 2000, XP and 2003. If you use Windows Vista, the installer displays the SSO Credential Provider as the authentication option.

Custom Configuration Files

You can push your custom configuration files as part of your CA SSO Client installation to end-users. To do this, you must provide the customized or preconfigured Client.ini, Auth.ini and Logging.properties file, and place them in the same directory level as the setup.exe file. The installation automatically uses these preconfigured files in preference to the embedded install settings.

Note: In order to silently install the CA SSO Client with a specific locale, set the language selection accordingly in the pre-configured Client.ini file prior to running the setup. For example:

```
[Localization]

Locale=DEU
```

Design Your Server Sets

A server set is a CA SSO term for a group of related servers and authentication information. The CA SSO Client uses these server sets to decide which CA SSO Servers and authentication agents it should refer to.

Sever sets extend the fault tolerance and failover of the CA SSO Client functionality where it interacts with the CA SSO Servers and the authentication agents.

Sever sets also help users identify which servers to log onto because you can give server sets meaningful names which appear in the drop-down list on the CA SSO Client logon screen. For example you could name two server sets, “Logon at Home” and “Logon at Work”.

Note: You do not have to create server sets if you are installing CA SSO Client with Citrix Metaframe Server support.

How to Create a Server Set

You must create at least one server set for the CA SSO Client to refer to.

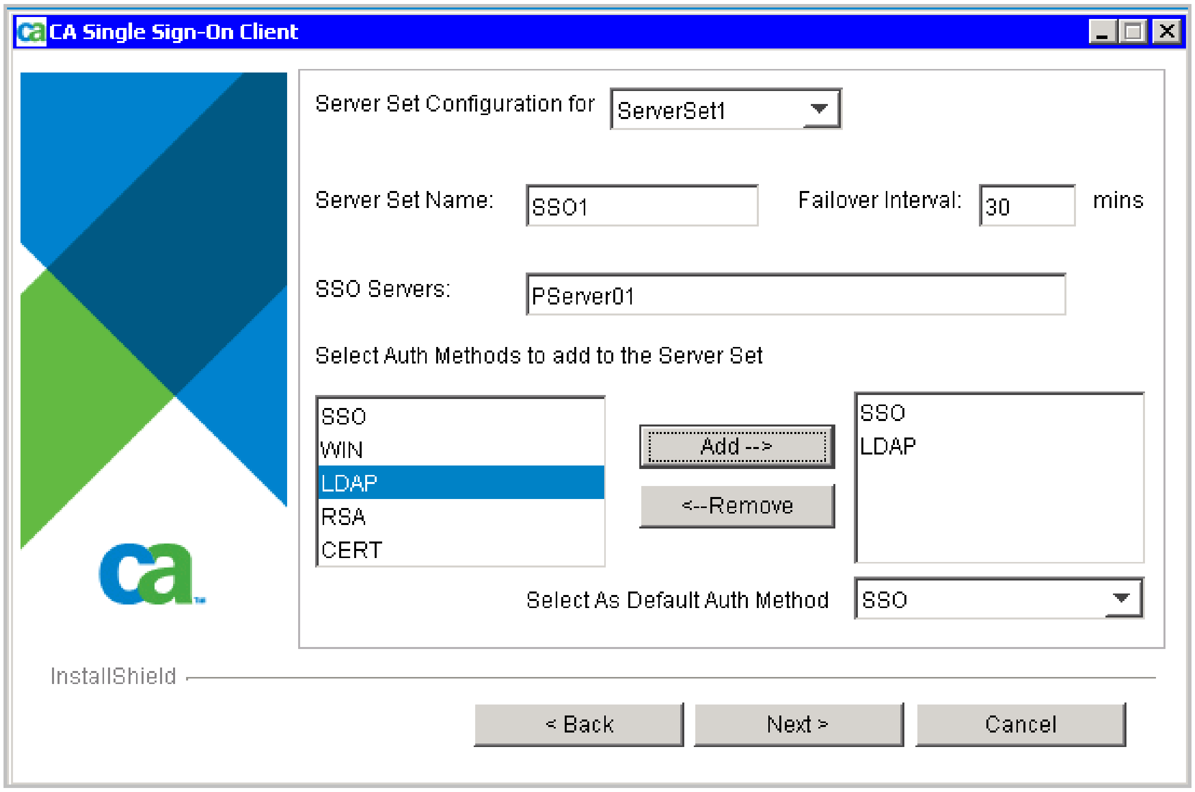
To create a server set you can either:

- Follow the CA SSO Client wizard installation (Typical or Custom)
- Edit the Auth.ini file using a text editor

  Note: You must install the CA SSO Client at least once to create an Auth.ini file.

How to Configure Server Sets

During the CA SSO Client installation, the following dialog appears to help you configure your server sets:



Server Set Configuration for: ServerSet1

Server Set Name: SSO1

Failover Interval: 30 mins

SSO Servers: PServer01

Select Auth Methods to add to the Server Set

SSO WIN LDAP RSA CERT	SSO LDAP

Select As Default Auth Method: SSO

Server Set Name:

Enter the name of the sever set. Make this a user-friendly name because users see this in their CA SSO Client logon screen list.

CA SSO Servers:

Enter the names of the CA SSO Server computers. You can enter multiple computer names separated by commas or white spaces. If the CA SSO Client cannot connect to the first computer in the list, it tries the second, and so on.

Failover Interval:

Enter the time that you want to elapse before the CA SSO Client retries a failed connection to a CA SSO Server. For example, if the CA SSO Client cannot connect to server_01 it tries server_02 and does not try to connect to server_01 again for 30 minutes.

Authentication Methods

Specify one or more authentication Method to be used by the server set.

Authentication Agent Servers:

Highlight an authentication method and specify the name(s) of the computer(s) that host(s) the authentication agent for the corresponding authentication software. You can enter multiple computer names separated by commas. If the CA SSO Client cannot connect to the first computer in the list, it tries the second, and so on.

Note: You must specify an authentication agent server for every authentication method that you want users to be able to use. The only exception is the built-in SSO authentication method.

After the first CA SSO Client is installed, you can copy and modify Server Set configuration information in the Auth.ini file to create a new server set.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the CA SSO Client:

- Ensure you meet all system requirements before you install the CA SSO Client. For information about supported platforms, see the *SSO Readme* file.
- Ensure you have your server set information ready. For each server set you need the names of the:
  - Server set (this is the name you create that users see in the authentication dialog)
  - Authentication agent server computers
  - CA SSO Server computers

- Ensure you know which authentication methods you want to use, for example, LDAP, SSO, RSA SecureID, Certificate, or Windows.
  - For Certificate authentication, specify the authentication methods, for example, PKCS#12, MSCAPI or PKCS#11. If you use PKCS#11, specify the PKCS#11 library.
  - For Windows authentication, you need to specify:
    - Trust file
    - Identity file and password (optional)
- Write and configure SSO scripts to logon to applications. You can install the CA SSO Client without these scripts, but the single sign-on functionality of launching applications relies on them.
- If you are upgrading the CA SSO Client from an earlier version, you receive an additional wizard dialog on migrating server set information.
- If you plan to install the CA SSO Client using silent installation, decide whether to use a response file or command line options.
- If you plan to use the SSO GINA:
  - Select custom during the graphical wizard installation process.
  - Ensure that you can run the installation using administrator privileges for the computer on which you install the CA SSO Client.
  - Create your own GINA dialog images if you do not plan to use the default SSO GINA dialog images.
- If you plan to use CITRIX, select the custom installation type during the graphical wizard installation process.
- If you plan to use FIPS communication mode between SSO components, you will need:
  - Identity file (in .pem format)
  - Private Key file (in .key format)

  Note: You can use sample certificates provided by the installer, or you can provide your own certificates during installation.

More information:

Implementing Authentication (see page 241)
Upgrade the CA SSO Client (see page 452)
Implementing Citrix Application Migration (see page 399)

setup Command—Install CA SSO Client

The command line options for installing the CA SSO Client include the following options:

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-P CitrixICAClientFeature.active

Specifies whether to install the SSO Client with Citrix ICA Client support. Set to True to specify installing SSO Client with Citrix ICA Client support.

Note: If set to 'true', Citrix ICA Client must be found on the computer otherwise installation will be aborted.

The command has the following format:

-P CitrixICAClientFeature.active=[*Value*]

Value: true | false

Default: false for Typical installs. Select true for Custom installs.

-P installLocation

Defines the install location.

The command has the following format:

-P installLocation=[*Value*]

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-P GinaFeature.active

Specifies whether to install SSO GINA and workstation lock mode.

The command has the following format:

-P GinaFeature.active=[*Value*]

Value: true | false

Default: false for Typical installs and true for Custom installs.

-P GinaPassThroughFeature.active

Specifies whether to install the SSO GINA Pass Through feature.

The command has the following format:

```
-P GinaPassThroughFeature.active=[Value]
```

Value: true | false.

Default: false for Typical installs and false for Custom installs.

-P CPFeature.active

Specifies whether to install SSO Credential Providers.

The command has the following format:

```
-P CPFeature.active=[value]
```

Value: true | false

Default: false

Note: This feature is supported only on Vista.

-V CP_STATIONLOCKMODE

Specifies the chosen Credential Provider lock workstation mode.

The command has the following format:

```
-V CP_STATIONLOCKMODE=[Value]
```

Value: 0 | 1 | 2 | 3 | 4 | 5

- 0= Single user station lock mode
- 1 = Multiple SSO users, single Windows user station lock mode
- 2 = Multiple WIN users, multiple Windows users station lock mode
- 3 = Multiple SSO users, no Windows user needed station lock mode (Kiosk mode)
- 4 = Single SSO User per Windows Session Mode
- 5 = Multiple SSO Users per Windows Session Mode

Default: 1

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

```
-V IS_REBOOT_NOW=[Value]
```

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V IS_SELECTED_INSTALLATION_TYPE

Specifies a typical or custom install.

The command has the following format:

```
-V IS_SELECTED_INSTALLATION_TYPE=[Value]
```

Value: Set to typical or custom.

Default: typical.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

```
-V LICENSE_VIEWED=[Value]
```

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V AUTH_METHODS

Defines the list of authentication methods for one server set. That is, one or more of: SSO, Certificate (CERT), LDAP, Windows (WIN) and RSA SecurID (RSA).

The command has the following format:

```
-V AUTH_METHODS=[Value]
```

Value: The list of authentication methods, for example, SSO, CERT, WIN, LDAP and RSA.

Note: Not required if installing SSO Client with the Citrix MetaFrame Server support.

-V DEFAULT_AUTHMETHOD

Defines the default authentication method for the user.

The command has the following format:

-V DEFAULT_AUTHMETHOD=[*Value*]

Value: The default authentication method, for example, SSO, CERT, WIN, LDAP and RSA. The values can be separated by a comma or space and must be surrounded by quotation marks if it contains space.

-V AUTHLDAP_AGENT_SERVERS

Defines the LDAP authentication agent server(s) name(s).

The command has the following format:

-V AUTHLDAP_AGENT_SERVERS=[*Value*]

Value: The list of servers hosting the LDAP authentication agent. List values can be separated by a comma or space and must be surrounded by quotation marks if they contain any spaces.

-V AUTHWIN_AGENT_SERVERS

Defines the Windows authentication agent server(s) name(s).

The command has the following format:

-V AUTHWIN_AGENT_SERVERS=[*Value*]

Value: The servers hosting the Windows authentication agent. List values can be separated by a comma or space and must be surrounded by quotation marks if they contain any spaces.

-V AUTHCERT_AGENT_SERVERS

Defines the CERT authentication agent server(s) name(s).

The command has the following format:

-V AUTHCERT_AGENT_SERVERS=[*Value*]

Value: The servers hosting the Certificate authentication agent. The values can be separated by a comma or space and must be surrounded by quotation marks if it contains space.

-V AUTHRSA_AGENT_SERVERS

Defines the RSA authentication agent server(s) name(s).

The command has the following format:

-V AUTHRSA_AGENT_SERVERS=[Value]

Value: The server/s hosting the RSA authentication agent. The values can be separated by a comma or space and must be surrounded by quotation marks if it contains space.

-V CERTAUTHMETHOD_PKCS12FILE

Defines the PKCS#12 file as one of the Certificate authentication method's certificate sources.

The command has the following format:

-V CERTAUTHMETHOD_PKCS12FILE=[Value]

Set to true to specify the PKCS#12 file as one of the Certificate authentication method's certificate sources.

Value: true | false

-V CERTAUTHMETHOD_MSCAPI

Defines MSCAPI as one of the Certificate authentication method's certificate sources.

The command has the following format:

-V CERTAUTHMETHOD_MSCAPI=[*Value*]

Set to true to specify MSCAPI as one of the Certificate authentication method's certificate sources.

Value: true | false

-V CERTAUTHMETHOD_PKCS11TOKEN

Specifies the PKCS#11 token as one of the Certificate authentication method's certificate sources.

The command has the following format:

-V CERTAUTHMETHOD_PKCS11TOKEN=[*Value*]

Set to true to specify the PKCS#11Token as one of the Certificate authentication method's certificate sources.

Value: true | false

-V CERTAUTHMETHOD_PKCS11LIB

Defines the PKCS#11 library file.

The command has the following format:

-V CERTAUTHMETHOD_PKCS11LIB=[*Value*]

Value: The PKCS#11 library file.

-V CITRIX_SERVER_SUPPORT

Specifies Citrix Metaframe Server support.

Set to true to specify installing SSO Client with Citrix MetaFrame Server support.

Note: If set to true, Citrix MetaFrame Server must be found on the computer otherwise installation will be aborted.

The command has the following format:

-V CITRIX_SERVER_SUPPORT=[*Value*]

Value: true or false

Default: false

-V GINA_STATIONLOCKMODE

Specifies the chosen GINA lock workstation mode.

The command has the following format:

-V GINA_STATIONLOCKMODE=[*Value*]

Value: 0 | 1 | 2 | 3

- 0= Single user station lock mode
- 1 = Multiple SSO users, single Windows user station lock mode
- 2 = Multiple WIN users, multiple Windows users station lock mode
- 3 = Multiple SSO users, no Windows user needed station lock mode (Kiosk mode)

Default: 1

-V FAILOVER_INTERVAL

Specifies the failover interval in minutes.

The command has the following format:

-V FAILOVER_INTERVAL=[*Value*]

Value: Value in minutes.

-V SSO_SERVERS

Defines the SSO Server name(s).

The command has the following format:

-V SSO_SERVERS=[*Value*]

Value: The SSO Server machine(s). The values can be separated by a comma or space and must be surrounded by quotation marks if it contains space.

-V SERVERSET_NAME

Defines the name of the server set.

The command has the following format:

-V SERVERSET_NAME=[*Value*]

Value: The name of the server set.

-V CLIENT_USERINTERFACE_CHOICE

Specifies the chosen SSO Client interface.

The command has the following format:

-V CLIENT_USERINTERFACE_CHOICE=[*Value*]

Value: 1 = SSO Launchbar | 2 = SSO Tools

Default: 1

-V CLIENT_USERINTERFACE_SHORTCUT_CHOICE

Specifies the location of the additional shortcut to the chosen SSO Client interface shortcut.

The command has the following format:

-V CLIENT_USERINTERFACE_SHORTCUT_CHOICE=[*Value*]

Value: 0 | 1 | 2

- 0=None
- 1=Desktop
- 2=Startup Group

Default: 1

-V CLIENT_FIPS_IDENTITY_FILE

Defines the Certificate file (.pem file) path for TLS communication between CA SSO Client and CA SSO Server.

The command has the following format:

-V CLIENT_FIPS_IDENTITY_FILE=C:\abc\cert.pem

Note: This option is required if COMM_MODE is set to 1 or 2.

-V WIN_AUTHMETHOD_TRUSTFILE

(Optional) Defines the trust file for the Windows authentication method.

The command has the following format:

```
-V WINAUTHMETHOD_TRUSTFILE=[Value]
```

Value: The WIN authentication method Trust file. Usually has a .pem extension.

-V WIN_AUTHMETHOD_IDENTITYFILE

Defines the Identity file for the WIN authentication method. This is optional.

The command has the following format:

```
-V WINAUTHMETHOD_IDENTITYFILE=[Value]
```

Value: The WIN AuthMethod Identity file. Usually has a .p12 extension

-V WIN_AUTHMETHOD_IDENTITYPASSWORD

Defines the password associated with the Identity file for the WIN authentication method.

The command has the following format:

```
-V WINAUTHMETHOD_IDENTITYPASSWORD=[Value]
```

Value: Password

-V VAR_WIN_AUTHMETHOD_FIPS_ID_PEM_FILE

Defines Identity File to use with Win Auth authentication method. This file is used in FIPS only mode and TLS modes of communication. The Identity file must be in a .pem format. If the value for this parameter contains spaces, the value must be enclosed in double quotes (" "). This parameter can be omitted if you use Custom Configuration Files.

```
-V VAR_WIN_AUTHMETHOD_FIPS_ID_PEM_FILE =[Value]
```

Value: Absolute path to the file of the Identity File including the file name.

-V VAR_WIN_AUTHMETHOD_FIPS_KEY_PEM_FILE

Defines WIN Auth Method private key file. This file is used in both FIPS-only and TLS modes of communication. Th private key file must be .key format. If the value for this parameter contains spaces, enclose it in double quotes (" "). This parameter can be omitted if you use Custom Configuration Files.

```
-V   VAR_WIN_AUTHMETHOD_FIPS_KEY_PEM_FILE =[Value]
```

Value: Absolute path to the private key file including the file name.

-V VAR_WIN_AUTHMETHOD_FIPS_TRUST_PEM_FILE

Defines WIN Auth Method trust file (mandatory for both FIPS only mode and TLS mode). Should be in .pem format. If the value for this parameter contains spaces, enclose it in double quotes (" "). This parameter can be omitted if you use Custom Configuration Files.

```
-V  VAR_WIN_AUTHMETHOD_FIPS_TRUST_PEM_FILE =[Value]
```

Value: Absolute path of the trust file including the file name.

-V MIGRATE_SERVERSET

Defines whether to migrate the Server Set settings during an upgrade. When set to "True", no Server Set configuration is required.

The command has the following format:

```
-V MIGRATE_SERVERSET=[Value]
```

Value: true | false

Set to true to migrate the Server Set settings during an upgrade.

Default: true

-V COMM_MODE

Defines the communication mode of the client.

The command has the following format:

```
-V COMM_MODE=[Value]
```

Value: 0 | 1| 2

- 0 - Compatible
- 1 - FIPS-only
- 2 - TLS mode

Default: 0

Install the CA SSO Client Using the Wizard

This topic explains how to install the CA SSO Client using the Product Explorer Wizard. You should use this method to install the CA SSO Client on individual computers.

Note: When you enter more than one computer name in a list, such as multiple CA SSO Servers, you can separate the names using either a comma or a space.

To install the CA SSO Client using the wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Client.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install the CA SSO Client Silently

You can install the CA SSO Client silently. You provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the CA SSO Client at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select CA SSO Client.
3. Click Install and follow the prompts until the license agreement is displayed.
4. Read the license agreement and note the license agreement command line setting. You need this setting to complete the silent install to show you have accepted the agreement. This is located at the bottom of the license agreement.
5. Click Cancel to exit the wizard and proceed with the silent installation.

   You can now install the CA SSO Client using silent installation.

6. Open a command prompt and navigate to the Client folder on the product DVD.
7. From the command prompt, type:

   ```
   setup.exe -silent -V LICENSE_VIEWED=value {parameters}
   ```

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=*value*

   Specifies whether you have viewed the license agreement found in the product install wizard.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

Install the CA SSO Client Silently using a Response File

Note: You can use a combination of response file and command line options to silently install the CA SSO Client. In this case, the command line options override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the Client folder on the product DVD.
3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create an CA SSO Client Response File (see page 144)
setup Command—Install CA SSO Server (see page 97)

Create an CA SSO Client Response File

Response files record user specific install information and are commonly used in silent installations. Response files can be created using the command *setup.exe -options-record (file name)*. The response filename can be a full path filename. This command launches the wizard install process, and all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the CA SSO Client folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record {file name}
   ```

   -options-record file name

   Specifies that the installer should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   A response file is created in the directory specified. If only the file name is specified, the response file is created in the CA SSO Client Folder.

Deploy the CA SSO Client using Unicenter Software Delivery (USD)

You can deploy the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD). To deploy the CA SSO Client and Session Administrator using USD, you need to:

1. Register the software install package with USD.
2. Configure install options.
   - Modify command line options
   - Modify response file options
3. Configure uninstall options.
4. Deploy the software to end users.

Note: The following procedures assume you have USD installed and operational.

1. Register the Software Installer Package with USD

To deploy SSO software using Unicenter Software Delivery (USD), register the software package with USD. Once registered, you need to configure install options prior to deploying to end users.

For more information on registering software with USD, see the *Unicenter Software Delivery Online Help*.

To register the software package with USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Right-click on Software Library and select Register, SD-Package.
3. On the Register SD Package dialog, click Browse and navigate to the appropriate install folder.

   Note: Select the folder that contains the install program, not the file itself.
4. Click Choose and then OK.
5. The software is copied and registered to USD.

2. Modify the Install Package Using USD

Use Unicenter Software Delivery (USD) to modify the software package install options. You can modify:

- Command line options
- Response file options

For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

Modify the Command Line Options

To modify command line options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software package you want to modify.
3. Right-click the install package and click Unseal.
4. Click the sub folders: Procedures, Install.
5. Select the software package and then right-click and select Properties.

6. Click each tab and make the required changes then click OK.

   Note: Ensure all command line information is correct in the Parameters field on the Embedded File tab. For more information on the command line, see the relevant silent install procedure in this chapter.

7. Right-click the software package and select Seal.

   The software package is ready for deployment.

   For more information on modifying install options using USD, see the *Unicenter Software Delivery Online Help*.

Important: You must correctly set the LICENSE_VIEWED parameter to indicate that you agree with the license agreement, otherwise the installation fails.

Modify Response File Options

Use the following procedure to modify the response file options.

To modify response file options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to modify.
3. Right-click the software package and click Unseal.
4. Click the sub folders: Procedures, Install.
5. In the right side panel, right-click the response file and select Properties.
6. Make the required changes then click OK.

   Note: For more information on response file install options, see the relevant setup command options in this chapter.

7. Right-click the software package and select Seal.

   The software package is ready for deployment.

   For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

3. Modify the Uninstall Package Information Using USD

Use Unicenter Software Delivery (USD) to modify the software package uninstall options. You can modify the uninstall command line information.

For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

To modify command line options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software package you want to modify.
3. Right-click the install package and click Unseal.
4. Click the sub folders: Procedures, Uninstall.
5. Select the software package and then right-click and select Properties.
6. Click each tab and make the required changes then click OK.

   Note: Ensure all command line information is correct in the Parameters field on the Embedded File tab.
7. Right-click the software package and select Seal.

   The software package is ready for deployment.

More information:

Uninstall Using Unicenter Software Delivery (USD) (see page 148)

4. Deliver the Software

To deploy SSO software using Unicenter Software Delivery (USD), you must first register the software with Unicenter Software Delivery.

The following procedure guides you through deploying software to a single user using USD. For more information on registering and deploying software using USD, see the *Unicenter Software Delivery Online Help*.

To deploy software to an end user

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and select your software package.

3. Right-click the install package and select Copy.
4. Click All Computers and Users.
5. In the right hand panel, select the computer you want to deploy the software to.
6. Right-click and select Paste>Software/Procedures to Schedule Jobs with Default Settings.

   A job container is created under the side menu heading of Job Containers.

   Note: For more deployment options, see the *Unicenter Software Delivery Online Help*.

Uninstall Using Unicenter Software Delivery (USD)

You can uninstall the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD).

Note: The following procedures assume you have USD installed and operational.

To uninstall using USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to uninstall.
3. Click the sub folder Installations.
4. Right-click All Computers and Users, Schedule Configure Job, Uninstall.

   Note: For more uninstall options, see the *Unicenter Software Delivery Online Help*.

Post-Installation Configuration Options

The topics that follow describe post-installation tasks.

Add CA SSO Client Features

You can add CA SSO Client features post-installation using the CA SSO Client installer. You can add the:

- GINA feature (Windows XP/2000/2003)
- GINA Pass Through features
- Citrix ICA Client feature
- Credential Provider feature (Windows Vista)

Note: For silent installation, you can use the command *setup.exe -silent -V LICENSE_VIEWED=value {parameters}* to specify the feature you want to install.

To add CA SSO Client features post-installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Client.
3. Click Install and select Custom on the Setup Type dialog.
4. Select the feature you would like to install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Remove CA SSO Client Features

You can remove individual CA SSO Client features using the CA SSO Client uninstaller using Add/Remove Programs in the Control Panel. You can uninstall the following individual components:

- GINA feature
- GINA Pass Through features
- Citrix ICA Client feature
- Credential Provider feature (Microsoft Windows Vista)

Note: For silent installation, you can use the command *setup.exe -silent -V LICENSE_VIEWED=value {parameters}* to specify the feature you want to install.

To remove CA SSO Client features post-installation

1. Click Start, Settings, Control Panel.
2. Select Add/Remove Programs.
3. Select CA Single Sign-On Client.
4. Click Change/Remove.
5. Select the features you want to remove.
6. Follow the prompts to un-install the feature.

Remove CA SSO Client Features Using Silent Uninstall

You can remove individual CA SSO Client features using the silent uninstall command: *<InstallDir>_uninst\uninstaller.exe –silent –options uninstall_rsp.txt*.

Where the uninstall_rsp.txt is used to identify the features you want to remove. For example, if you want to remove the Citrix ICA Client feature, you would ensure the uninstall_rsp.txt file contains the following:

```
-P CitrixICAClientFeature.activeForUninstall=true
```

Note: If you want to retain certain CA SSO Client features, ensure they are set to False, otherwise they will be removed during the uninstall.

Important: If you set *-P SSOClientFeature.activeForUninstall=true*, then the whole product is uninstalled.

Repair CA SSO Client Features

You can repair existing CA SSO Client features post-installation using the CA SSO Client installer.

To repair CA SSO Client features post-installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer Wizard automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Client.
3. Click Install.

   The pre-existing features installed are selected.

4. Click "Next" and follow the prompts to repair the feature/s.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Configuring the CA SSO Client

If you need to repair or modify CA SSO Client components, you can use the:

- Installation wizard from the SSO DVD, as you did for the CA SSO Client installation and click Modify.

  The install wizard detects that the CA SSO Client is installed and shows the appropriate interface for repair and/or modify-by-adding of CA SSO Client components.

- Add/Remove Programs Panel from the Windows Start menu, and select CA SSO Client and click Remove.

  The uninstall wizard shows the appropriate interface for remove and/or modify-by-removing of CA SSO Client components.

You can also modify the CA SSO Client by manually editing the configuration files:

- Auth.ini file
- Client.ini file
- Logging.properties

Note: We recommend that you shut down the CA SSO Client (including SSO Tools, SSO Status Icon and SSO Launchbar) before you make any changes to the configuration files. If you do not shut down the CA SSO Client, you must restart it for changes to take effect.

Central CA SSO Client Configuration

The behavior of the CA SSO Client is controlled by the CA SSO Client configuration files (Client.ini and Auth.ini). These files are stored on each CA SSO Client computer and control the behavior of the CA SSO Client that computer. You can configure these files to automatically check a central server for new configuration files using a setting in the Client.ini file.

The CA SSO Client can detect an updated ini file on the central server, pull it down and apply the changes ‘in flight’. This means that you do not need to restart the CA SSO Client if making changes this way to apply them.

Configure a Central CA SSO Client Configuration File

You can configure each Client.ini and Auth.ini to periodically update itself from a central server. This means that you can regularly update the entire configuration for CA SSO Clients quickly and easily.

To configure a regular update of the CA SSO Client configuration file from a central location

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Navigate to the [ConfigurationSource] section.

3. Edit the following values:

   ClientIniFile

   Defines the location on the network of the central Client.ini file.

   Value: *path and file name*

   Default: [no default]

   AuthIniFile

   Defines the location on the network of the central Auth.ini file.

   Value: *path and file name*

   Default: [no default]

   CachePeriod

   Defines how frequently the SSO Client should download the central SSO Client configuration files (Client.ini and Auth.ini) from the central network location.

   Value: *time in [days]d[hours]h[minutes]m[seconds]s*

   Default: 1d

4. Save the Client.ini file.

   Note: We recommend that you shut down the CA SSO Client (including SSO Tools, SSO Status Icon and SSO Launchbar) before you make any changes to the configuration files (Client.ini). If you do not shut down the CA SSO Client98, you must restart it for changes to take effect.

More Information

Auth.ini Configuration (see page 153)
Client.ini Configuration (see page 163)
Registry Configuration (see page 188)

Auth.ini Configuration

This section of the Auth.ini file defines settings that affect the authentication options.

[AuthOptions]

ServerSetSelection

Defines whether the user can select a server set when they click the Change button on their authentication dialog.

0 = Always show the server set selection to the user

1 = Only show the server set selection if more than one server set or authentication method is assigned to the user.

2 = Automatically selects the server set and authentication method that was selected by the pervious user, if the RememberLastUser token is activated in this file. If there is only one server set and authentication method assigned to the user, they are not prompted to select these.

3 = Never show server set selection to the user

Value: [0|1|2|3]

Default: 0

Notes:

Login Process

- The ServerSetSelection parameter controls the display of login, lock, and unlock screens that are displayed to users. If ServerSetSelection is set to 0, 1 or 2, the Server Set dialog is displayed to users and users can change the authentication method and domain used for authentication.
- If ServerSet Selection is set to 3, the login screen populates the default authentication method and domain and prompts the user for username and password. Users can change the authentication method and domain by pressing ESC.

Lock and unlock

- If ServerSetSelection is set to 0, 1 or 2, the Server Set dialog is displayed to users and users can change the authentication method and domain used for authentication.
- If ServerSetSelection is set to 3, displays the locked tile of the previous user, to changed username you must press ESC. The user tile then populates the authentication methods and domain of the previous user and prompts for username and password. Users can change the authentication method and domain by pressing ESC.

RememberLastUser

Defines whether the Client remembers the previous user who logged on and uses those details to pre-populate the server set dialog during the SSO logon.

Value: [yes|no]

Default: yes

ServerSetAlphabetic

Defines whether to list the server sets alphabetically. If set to no, the Client will order these according to server set number.

Value: [yes|no]

Default: no

RestrictLogonDomain

Defines whether to restrict the user's access to the domain. If this is set to yes, the user cannot select the domain from the SSO GINA window. They will only be able to select to logon to their own computer, without access to the domain.

Value: [yes|no]

Default: [no default]

This section of the Auth.ini file defines settings that affect the server sets.

[ServerSet1]

Name

Defines the name that you want users to see in their Server Set drop-down list on the authentication screen. For example, "Home Logon" or "Work Logon".

Value: *server set name*

Default: [None]

PolicyServers

Defines the list of SSO Servers in a set. Use spaces to separate. When an SSO Client attempts to connect to a SSO Server, it tries to use the first SSO Server in the list. If that SSO Server is not available, the second SSO Server in the list is used. For example: server1 server2. You must never leave this value empty.

Value: *server name(s)*

Default: [no default]

AuthMethods

Defines the list of authentication methods available to users. Use spaces to separate. The first value listed here appears in order in the Authentication dialog and the first value is the default. If the first value fails to load, the next value is used.

You must specify an authentication host in the AuthXXX section below that corresponds to the authentication method(s) defined here or the Server Set is deemed invalid and will not be displayed.

The exceptions to this rule are the AuthWIN and AuthCITRIX methods that will allow the use of the keyword <auto>. The <auto> keyword signifies to locate the nearest domain controller, or nearest Citrix server respectively.

For example:

```
AuthMethods=LDAP
AuthLDAP=AuthServer1 AuthServer2
```

Value: [CERT|LDAP|RSA|SSO|WIN|

Default: [See description]

OfflineTimeout

Defines the time in seconds that elapses before an authentication host that has been marked as offline can be retried. If all the hosts specified for a given ServerSet/authentication method are marked as offline, rather than making the authentication attempt fail, all associated hosts will be retried again.

The minimum value is 300 seconds (5 minutes). If you set this value to less than 300, it will automatically adjust this value to 300.

Value: *time in seconds*

Default: 1800

LogicalAuthSSO

Defines the logical authentication hosts to use for SSO authentication.The default value is SSO_Authhost which is automatically configured during installation.

Note: If you want to edit the Auth.ini file post installation, the LogicalAuthSSO setting must always be set to SSO_Authhost. Any other value could cause SSO authentication to fail.

Value: SSO_Authhost

Default: SSO_Authhost

AuthLDAP

Specify the list of the authentication hosts to use for LDAP authentication. Use spaces to separate.

There is no default value for AuthLDAP unless specified during the installation.

When an SSO Client attempts to authenticate, it tries to use the first host in the list. If that host is not available, the next host in the list is used.

Value: *server name(s)*

Default: [See description]

AuthWIN

Specify the list of the authentication hosts to use for Windows authentication. Use spaces to separate.

There is no default value for AuthWIN unless specified during the installation.

When an SSO Client attempts to authenticate, it tries to use the first host in the list. If that host is not available, the next host in the list is used.

Value: *server name(s)*

Default: [See description]

AuthCERT

Specify the list of the authentication hosts to use for Certificate authentication. When an SSO Client attempts to authenticate, it tries to use the first host in the list. If that host is not available, it uses the second host in the list and so on.

There is no default value for AuthCERT unless specified during the installation.

Value: *server name(s)*

Default: [See Description]

AuthRSA

Specify the list of the authentication hosts to use for RSA SecureID authentication (previously referred to as SDI authentication). Use spaces to separate.

There is no default value for AuthRSA unless specified during the installation.

When an SSO Client attempts to authenticate, it tries to use the first host in the list. If that host is not available, the next host in the list is used.

Value: *server name(s)*

Default: [See description]

AuthCITRIX

Specify the Citrix Server to use.

The <auto> value signifies that the Citrix virtual channel will be used to retrieve the user's SSO ticket from the user's local computer and for that ticket to be used automatically by the SSO Client on the Citrix Server.

Value: *Citrix server name(s)*

Default: <auto>

[Auth.WIN]

ConnectionTimeout

Defines how long in seconds the SSO Client tries to connect to the Windows authentication server before it times out.

Value: *number of seconds*

Default: [no default]

IdentityFile

Defines the certificate file on the SSO Client computer. This certificate file and password are supplied during the SSO Client installation. This ensures an SSL-secured connection between the SSO Client computer and the Windows authentication agent computer.

Value: *file name*

Default: Supplied during SSO Client installation

IdentityPassword

Defines the password associated with the certificate file (defined by the IdentityFile token). This certificate file and password are supplied during the SSO Client installation. This ensures an SSL-secured connection between the SSO Client computer and the Windows authentication agent computer. To alter this value manually at a later stage, use the Obfuscation Tool ‘ssoencconf.exe’ supplied with the product. You can find this tool in the bin directory for this component.

Value: *password*

Default**: Supplied during SSO Client installation**

TrustFile

Defines the .pem file that establishes trusted communications between the SSO Client computer and the Windows authentication computer. This is mandatory. This value is entered during installation of the SSO Client.

Value: *file with .pem extension*

Default: Supplied during SSO Client installation

FIPSIdentityFile

Defines the certificate .pem file used for TLS communications between the CA SSO Client computer and the Windows authentication computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .pem extension*

Default: Supplied during CA SSO Client installation in FIPS or Mixed Mode.

FIPSTrustFile

Defines the .pem file that establishes trusted communications between the CA SSO Client computer and the Windows authentication computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .pem extension*

Default: Supplied during CA SSO Client installation in FIPS or Mixed Mode.

PrivateKeyPath

Defines the certificate .key file used for TLS communications between the CA SSO Client computer and the Windows authentication computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .key extension*

Default: Supplied during CA SSO Client installation in FIPS or Mixed Mode.

AutoNetworkAuth

Defines whether to let the Windows authentication method use the user's network credentials to log them on to the SSO Client.

If no, the user must enter their credentials manually.

If yes, we recommend that the SSO administrator adds a logoff script to log the authenticated user off the network.

Value: [yes|no]

Default: No

PDCFallback

Defines whether Windows authentication attempts to find an authentication host on the domain if none of the specified hosts are online or the specified host is '<auto>'.

Setting the host to '<auto>' and PDCFallback to 'no' is invalid and causes Windows authentication to fail.

Value: [yes|no]

Default: yes

NearestDomainController

If none of the specified hosts are online, or the specified host is '<auto>', Windows authentication attempts to find an authentication host on the domain.

In this case, this value is used to define whether the SSO Client must try to authenticate to the nearest Domain Controller (DC).

If yes, the SSO Client tries to connect to the nearest DC on the network regardless of the target OS (Active Directory architecture).

If not, the SSO Client tries to connect to the Primary Domain Controller (PDC) specified.

If <auto>, the SSO Client checks the operating system and behave accordingly. It tries to connect to the nearest DC (presumes Active Directory architecture) and if this fails, connects to the PDC (presumes NT4 architecture).

This value does not have any effect if PDCFallback is set to no.

Value: [yes|no|<auto>]

Default: <auto>

[Auth.CERT]

CertStore

Defines where the user certificate is stored.

- FILE = Stores user certificate in a local disk file (PKCS#12 format)
- PKCS11 = Stores the user certificate on a PKCS#11 token (smart card in most cases).
- MSCAPI = Uses the MSCAPI certificate store or smart card

You can specify all storage methods, for example:

```
certStore=PKCS11 FILE
```

Value: [FILE|PKCS11|MSCAPI]

Default: FILE

Pkcs11LibraryPath

Defines the full path name of the PKCS#11 library that you want to use with the type of PKCS#11 token you have selected.

This must be defined otherwise you will not be able to use the PKCS#11 token. This property is only relevant if CertStore=PKCS11.

If the CertStore attribute is set to FILE, then the attribute can be left empty.

Value: *path*

Default: [no default]

Pkcs11PromptText

Defines the prompt the user sees when the PKCS#11 token radio button is selected on the certificate authentication page.

If this is left empty, the default text will be used. The default text is: “Insert your token into reader and enter your password.”

Value: *text*

Default: [See description]

DisablePasswordField

Defines whether to disable the password field. You might do this when you want to force an authentication method other than password or PIN.This is only related to authentication using PKCS#11 token.

If the password field is disabled, the system forces the user to use a third-party authentication method (such as user's fingerprints) to authenticate to the smart card.

Value: yes|no

Default: no

AutoAuthenticate

Defines whether to automatically authenticate the user using the certificate authentication method with the certificate stored in the Windows user's personal certificate store. To view the certificates in the Windows user’s personal store, open Internet Explorer and click Tools, Internet Options, Content, Certificates, Personal. This applies to the MS CAPI certificate store only. (CertStore=MSCAPI)

Value: [yes|no]

Default: no

CertThumbPrint

Defines the thumb print (SHA1 hash) of the certificate in the Windows user's personal certificate store to use when carrying out automatic authentication (AutoAuthenticate=yes). This is only related to automatic authentication using MS CAPI (CertStore=MSCAPI). This setting is frequently used in Shared Computer environments. To get the thumb print of the certificate you want to use, open Internet Explorer and go to Tools, Internet Options, Content, Certificates, Personal. Select the certificate you want to use and click View. On the Details page of the dialog that pops up, there is a field named Thumbprint. Copy the value of this field and use it as the value of this configuration setting. Remember to remove any spaces among the text.

Value: text

Default: [no default]

DefaultServerSet

Defines which Server Set to use with Certificate authentication if the user starts the authentication process by inserting a PKCS#11 token or an MS CAPI-enabled smart card into one of the PKCS#11 slots or smart card readers that are connected to the end user workstation. This is only relevant when the end user is using SSO GINA to log on and the user is facing the GINA welcome dialog.

Value: *server set name*

Default: [no default]

IdentityFile

Defines the certificate .pem file used for TLS communications between the CA SSO Client computer and the Certificate authentication Agent computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .pem extension*

Default: Supplied during the CA SSO Client installation in FIPS or Mixed Modes.

[Auth.LDAP]

IdentityFile

Defines the certificate .pem file used for TLS communications between the CA SSO Client computer and the Certificate authentication Agent computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .pem extension*

Default: Supplied during the CA SSO Client installation in FIPS or Mixed Modes.

[Auth.RSA]

IdentityFile

Defines the certificate .pem file used for TLS communications between the CA SSO Client computer and the Certificate authentication Agent computer. This is mandatory. This value is entered during installation of the CA SSO Client in FIPS or Mixed Mode of installation.

Value: *file with .pem extension*

Default: Supplied during the CA SSO Client installation in FIPS or Mixed Modes.

[Auth.CITRIX]

ReadTimeOut

Defines the length of time the SSO Client on the Citirix Server waits for a response from the SSO Client on the end user workstation.

Value: *time*

Default: 60s

Client.ini Configuration

This section of the Client.ini file defines settings that affect the user experience.

[Localization]

Locale

Defines which language alphabet the SSO Client interface can display. This is indicated by a three letter language code. SSO is not formally localized for the r8.1 release, but it does support non-English characters.

Value: *three character language code*

DEU = German

ENU = English

ESP = Spanish

FRA = French

ITA = Italian

JAP = Japanese

Default: ENU

ServerMessageFile

Defines the file that contains error messages that the user sees when the Client encounters problems connecting to the SSO Server.

Value: *path and file name*

Default: %SSOINSTALLDIR%\res\enu.msg

HelpDeskMessage

Defines an additional message shown to users when the Client encounters an unexpected error. If this value is left blank, no additional message will be displayed to the user.

Value: *text*

Default: If problems persist, contact the person that manages your CA SSO account.

MessageOfTheDayFile

Defines the location of the message of the day file. This is a message that the user sees when they log onto CA SSO.

Value: *path and file*

Default: [no default]

[LaunchBar]

StartDocked

Defines whether the Launchbar starts docked to one edge of the screen, or whether it is free-floating. The dock location is defined by the DockedEdge token.

Value: [yes|no]

Default: no

DockedEdge

Defines which edge of the screen the Launchbar will start docked to if you specified StartDocked=yes.

Value: [top|bottom|left|right]

Default: top

OffSetX

Defines the X coordinate (horizontal plane) of the Launchbar from the top left corner of the dialog to the top left corner of the screen. This value only applies to the SSO Launchbar if it is in "floating" mode. If StartDocked=yes, this value is ignored.

Value: *The number of pixels*

Default: the window is centred

OffSetY

Defines the Y coordinate (vertical plane) of Launchbar from the top left corner of the dialog to the top left corner of the screen. This value only applies to the SSO Launchbar if it is in "floating" mode. If StartDocked=yes, this value is ignored.

Value: *number of pixels*

Default: the window is centred

AlwaysOnTop

Defines whether the Launchbar should always be visible and stay on top of all other windows. This value is only valid if StartDocked=no.

Value: [yes|no]

Default: no

AutoHide

Defines whether the Launchbar should automatically hide until the mouse moves over it. This token is only valid when the Launchbar=yes.

Value: [yes|no]

Default: no

AutoLogon

Defines whether to initiate the authentication sequence as soon as the Launchbar is started, as opposed to letting the user click the "Logon" button on the Launchbar.

Value: [yes|no]

Default: no

AppLineCount

Defines the default number of application icons that appear in each row of the Launchbar window.

Value: *Number of applications*

Default: 3

DisplayUserName

Defines whether the name of the user must appear in the window title.

Value: [yes|no]

Default: yes

RestorePosition

Defines whether the Launchbar must revert back to its original position when the user logs off.

Value: [yes|no]

Default: no

ConfirmLogoff

Defines whether the user sees a dialog asking them to confirm if they want to log off CA SSO.

Value: [yes|no]

Default: no

ButtonSize

Defines the size of the buttons on the Launchbar.

- Small (16x16 pixels)
- Medium (32x32 pixels)
- Large (48x48 pixels)
- Auto (Median size, but adjusts to the width of the icon to fit the longest caption. You can limit this using MaxCaptionLength)

Values: [small|medium|large|auto]

Default: auto

MaxCaptionLength

Defines the character limit on the width of the caption of any button that appears on the launchbar. This token is only valid when ButtonSize=auto.

Value: *number of characters*

Default: 12

DisplayLogonButton

Defines whether the user sees a Logon button on the Launchbar. When the user clicks this button they will be prompted to authenticate. If you set this to no, to remove the user's control over Logon/Logoff, you may also want to remove the Logon/Logoff buttons on all the user interfaces: the Status Icon and SSO Tools.

Value: [yes|no]

Default: yes

DisplayLockComputerButton

Defines whether the user sees a Lock Computer button on the Launchbar.

Value: [yes|no]

Default: yes

DisplayOptionsButton

Defines whether the user sees an Options button on the Launchbar. When the user clicks the Options button they will see options to:

- Exit the Launchbar
- Refresh their application list
- Display their details
- Auto hide the LaunchBar, if docked
- Display the LaunchBar on top of other windows

Value: [yes|no]

Default: yes

DefaultApplicationIconFile

Defines what icon the user sees for an application in the user interfaces. This is only used if you do not assign a custom icon for an application. If left blank, this will automatically use the default generic SSO application icon.

Value: *path and file name (.ico extension)*

Default: [no default] (therefore the SSO default icon is used)

DefaultApplicationIconIndex

Defines which icon to display to the user if there are multiple icons in the file defined by DefaultApplicationIconFile. If you set this to 0 the first icon will be used. If you set this to 1 the second icon will be used and so on.

Value: see description

Default: 0

[LaunchBar/OptionsMenu]

AlwaysOnTop

Defines whether the "Always on top" item appears in the Options menu on the Launchbar. This lets the user control the behavior of the SSO Launchbar and overrides the AlwaysOnTop token in the Launchbar section.

Value: [yes|no]

Default: yes

AutoHide

Defines whether the "Auto Hide" item appears in the Options menu on the Launchbar. This lets the user control the behavior of the SSO Launchbar and overwrites the AutoHide token in the Launchbar section.

Value: [yes|no]

Default: yes

RefreshApplist

Defines whether the "Refresh Application List" item appears in the Options menu on the Launchbar. This lets the user refresh their application list. You might set this value to no if you wanted to reduce the load on the servers in a large-scale implementation.

Value: [yes|no]

Default: yes

UserConfiguration

Defines whether the "My Details" item appears in the Options menu on the Launchbar. This lets the user change their authentication credentials, if the authentication method supports this.

Value: [yes|no]

Default: yes

Exit

Defines whether the Exit Launchbar item appears in the Options menu on the Launchbar. This lets the user exit from CA SSO.

Value: [yes|no]

Default: yes

[LaunchBar/AppMenu]

EnableAdvancedApplication

Defines whether the user sees a menu when they right-click on an application. This menu lets the user:

- Change their password
- Create a shortcut for that application on their desktop
- Configure the application to run as soon as they log onto Windows

Value: [yes|no]

Default: yes

[GINA]

LogonBitmap

Defines the name (including the path) of an image to use for the SSO GINA's logon window.

If this value is omitted, or that image cannot be loaded, the GINA uses a default bitmap which says “Welcome to CA Single Sign-On". You can customize this text using LogonText value.

Value: *path and image name*

Default: [no default]

LogonTitle

Defines the title for the SSO GINA's logon window. If not specified, the default value is “Windows Logon”.

Value: *dialog title*

Default: Windows Logon

LogonText

Defines the text the user will see in the SSO GINA Logon dialog.

Value: *Text*

Default: Welcome to CA Single Sign-On

LockedBitmap

Defines the name (including the path) of an image to use for the SSO GINA's 'Station locked' window.

If this value is omitted or that image cannot be loaded, the GINA uses a default SSO bitmap which says "This computer is in use and has been locked".

Value: *path and image*

Default: [no default]

LockedTitle

Defines the title for the SSO GINA's 'Station locked' window. If not specified the default value of “Computer Locked” is used.

Value: *dialog title*

Default: Computer Locked

LockedText

Defines the text that the user sees on the SSO GINA when the computer is locked.

Value: *text*

Default: This computer is in use and has been locked.

Font

Defines the font used for LogonText and LockedText on the GINA windows.

Value: *any system font by name*

Default: Arial

FontSize

Defines the size of the font specified in the Font value.

Value: *font size*

Default: 13

GinaPassThrough

Defines whether to bypass the SSO GINA and go to the Microsoft GINA (MSGINA), even if the SSO GINA is installed.

If GinaPassThrough is set to no, the user sees the SSO GINA in all cases (welcome screen, logon screen, Ctrl+Alt+Del options screen and locked screen).

If GinsPassThrough is set to yes, the user sees the MSGINA screen for the logon screen, but sees the SSO GINA for all other GINA screens.

You need to use this setting if you plan to use Full Shared Computer mode. This is used when more than one person shares a computer and each user can share a generic Windows setup, but each user needs to have their own customized SSO applications.

Value: [yes|no]

Default: **no**

LogonCAD

Specify whether to display the Ctrl + Alt + Del dialog at logon.

Value: [yes|no]

Default: yes

FetchDomainsFromSystem

Defines from where the Domains are fetched. If yes is specified, the Domains are fetched from the network. If set to no, the Domains are fetched from the Key Domains INI file.

Value: [yes|no]

Default: yes

Domains

This key is used only if the FetchDomainsFromSystem key is set to no.

Value: *list of domains separated by spaces.*

Default: [no default]

[GINA/SystemLogon]

NetWareLogon

Defines whether to use different Windows and NetWare logon credentials. If you set this value to 'yes' you will need to provide an SSO logon script for NetWare logon. This will perform netware/NDS logon when a user performs sign-on from the GINA.

Value: [yes|no]

Default: no

NetWareServer

Defines the default NetWare server name. This is only necessary if you set NetWareLogon token to 'yes' and the NetWare client does not already specify a preferred server in the following location: HKLM\System\CurrentControlSet\Services\NetwareWorkstation\Parameters\Preferred Server

Value: *Name of NetWare Server*

Default: [no default]

[GINA/StationLock]

EnableOsUnlock

Defines whether you want users to be able to unlock their computer using the Windows Logon option only on the SSO GINA. If this is set to no and the user cannot unlock the computer using their SSO credentials then they will not be able to revert to unlocking the computer using the Windows logon. (Users can override this setting and get the option to bypass SSO by pressing the CTRL + ALT + Z for a dialog instance).

Value: [yes|no]

Default: yes

DisableShutdown

Defines whether you want to disable the Shutdown button. The user sees the Shutdown button on the Windows Authentication dialog when they log on using the SSO GINA and on the Secure Information dialog when they press presses Ctrl + Alt + Del using the SSO GINA.

Value: [yes|no]

Default: no

DisableLogoff

Defines whether you want to disable the logoff button. The user sees the Logoff button on the Secure Information dialog when they press Ctrl + Alt + Del using the SSO GINA.

Value: [yes|no]

Default: no

UnlockStationMode

Defines the workstation mode. These values only apply when the SSO GINA and Credential Providers are in use. When GINA is in use, the valid values are zero through three. For Credential Providers the the valid values are zero through five.

- Single user lock option

  Select option 0. This is used in a regular non-shared computer environment (one Windows user, one SSO user).

- Multiple SSO user lock option
  Select option 1. This is used when two or more people share a customized Windows setup, but need access to their own SSO applications on a computer (one Windows user, multiple SSO users).

- Multiple Windows user lock option
  Select option 2. This is used when more than one person shares a computer and each user needs to have their customized Windows setup as well as their own SSO applications (multiple Windows users, multiple SSO users).

- Kiosk mode lock option
  Select option 3. This is used when more than one person shares a computer and shares a generic Windows setup, but each user needs to have their customized SSO applications. This is like option Multiple Windows user lock mode, but is much faster and suits an environment where several people may have to use one computer in quick succession or may not have a Windows user account.

- Single user Windows session lock option
  Select option 4. This is used in a non-shared Windows session environment (one CA SSO user for one Windows user, but multiple Windows sessions (accounts) are allowed)

- Multiple users Windows session lock option
  Select option 5. This is used when two or more people share the same Windows account but need access to their own SSO applications on a computer (multiple SSO users for one Windows users, but multiple sessions are allowed).

Value: [0|1|2|3|4|5]

Default: **1**

ShowLockedUsername

Defines whether you want the name of the user who locked the computer to appear on the title bar while the computer is locked.

Value: [yes|no]

Default: yes

[SSOTools]

DisplayLogonButton

Define whether the user sees the Logon/Logoff button on the SSO Tools dialog. If you set this to no, to remove the user's control over Logon/Logoff, you may also want to remove the Logon/Logoff buttons on all the user interfaces: the Launchbar and the Status Icon.

Value: [yes|no]

Default: yes

EnableChangePasswordButton

Define whether the "Change Password" button is enabled. This button lets users change their logon credentials for a specific application.

Value: [yes|no]

Default: yes

EnableRefreshButton

Defines whether the Refresh List button is enabled on the SSO Tools interface. This button lets users update their application list from the SSO Server. You might set this value to 'no' if you wanted to reduce the load on the servers in a large-scale implementation.

Value: [yes|no]

Default: yes

ExplorerOpenFolder

Defines whether the Open button is enabled on the SSO Tools dialog when an application folder is selected. If the user clicks this button it launches Windows Explorer and opens to the respective folder in their Start menu if the corresponding folder is created. This is related to the StartMenu_CreateLinksAutomatically value.

Value: [yes|no]

Default: no

[StatusIcon]

DisplayNotifications

Defines whether the user sees the status balloons that pop up from the SSO Status Icon. The two messages the user might see are "Connection to the SSO Server has been lost" and "Connection to the SSO Server has been restored".

Value: [yes|no]

Default: yes

[StatusIcon/Menu]

Logon

Define whether to include the Logon option in the Status Icon menu. If you set this to no, to remove the user's control over Logon/Logoff, you may also want to remove the Logon/Logoff buttons on all the user interfaces: the Launchbar and SSO Tools.

Value: [yes|no]

Default: yes

Logoff

Define whether to include the Logoff option in the Status Icon menu. If you set this to no, to remove the user's control over Logon/Logoff, you may also want to remove the Logon/Logoff buttons on all the user interfaces: the Launchbar and SSO Tools.

Value: [yes|no]

Default: yes

RefreshApplicationList

Define whether to include the Refresh Application List option in the Status Icon menu. When a user clicks this the SSO Server recalculates that user's application list and sends it to the SSO Client. You might set this value to 'no' if you wanted to reduce the load on the servers in a large-scale implementation.

Value: [yes|no]

Default: yes

OpenSSOTools

Define whether to include the Open SSO Tools option in the Status Icon menu. When a user clicks this, the SSO Tools window opens.

Value: [yes|no]

Default: yes

OpenLaunchBar

Define whether to include the OpenLaunchBar option in the Status Icon menu. When a user clicks this the SSO Launchbar opens.

Value: [yes|no]

Default: yes

LockComputer

Define whether to include the Lock Computer option in the Status Icon menu. When a user clicks this the workstation is locked.

Value: [yes|no]

Default: yes

ApplicationList

Define whether to include the ApplicationList option in the StatusIcon menu. When a user clicks this they will see their application list as a sub-menu in the StatusIcon menu.

Value: [yes|no]

Default: yes

AboutSSO

Define whether to include the About option in the Status Icon menu. When a user clicks this they will see system information about SSO such as the version number.

Value: [yes|no]

Default: yes

Exit

Define whether to include the Exit option in the Status Icon menu. When a user clicks this the SSO Client Status Icon closes. If the Status Icon is closed, restarting the workstation will restart it.

Value: [yes|no]

Default: yes

[ApplicationLinks]

StartMenu_Folder

Specify the name of the folder in the Start menu that displays the shortcuts to the SSO-supported applications.

Value: *item name in the Start menu*

Default: SSO Programs

StartMenu_Hierarchy

Defines whether the user sees their SSO-enabled applications in their application containers on their Windows Start menu. If this is set to 'no', the user will see all their SSO-enabled applications in a flat list.

Value: [yes|no]

Default: yes

StartMenu_CreateLinksAutomatically

Defines whether the user sees an SSO submenu in the Windows Start menu.

Value: [yes|no]

Default: yes

StartMenu_RemoveLinksAutomatically

Defines whether to delete the SSO submenu from the Windows Start menu (and all the links it contains) when the SSO Client is shut down. You would set this value to 'yes' if you had SSO set up in a Shared Computer environment in order to remove the previous user's shortcuts when a new user logs on.

Value: [yes|no]

Default: yes

AllowUserToCreate

Defines whether the user has the ability to create a shortcut on their Windows desktop or in their Startup Group (to start automatically when Windows starts). The user can configure these options using the check boxes on SSO Tools or by right-clicking an application in the Launchbar and selecting these options from the pop-up menu. You might choose to disable this option if you had SSO set up in a full or partially shared computer environment to avoid a large number of applications being started at Windows startup.

Value: [yes|no]

Default: yes

[UserPage]

EnableChangeAuthCredentials

Defines whether a user can change their primary credentials from the User window (accessed from the SSO Tools, My Details tab or from Launchbar, Options, My Details).

Value: [yes|no]

Default: yes

This section of the Client.ini file defines the settings that affect the core functionality.

[AppListRefresh]

Enabled

Defines whether you want to turn the SSO Client's automatic application list refresh on.

If this is set to 'no', the rest of the tokens in this section are ignored.

Value: [yes|no]

Default: no

Interval

Defines the time between checks for an updated application list.

Note: If this value is set, then the EarliestStartTime and LatestStartTime values are ignored.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: [no default]

EarliestStartTime

In conjunction with LatestStartTime, defines the time period within which a daily refresh occurs (random point between EarliestStartTime and LatestStartTime). You might want to schedule this during low-network traffic periods.

If these values are set, they are only used if the 'Interval' token is not set.

The time is specified as a 24 hour clock: for example, 21:31 indicates 9:31 pm.

Value: *time in [hours]:[minutes]*

Default: 09:00

LatestStartTime

For a full description, see EarliestStartTime in this section.

Value: *Time in [hours]:[minutes]*

Default: 17:00

[Cache]

CacheDirectory

Defines the location of the cache directory that stores the SSO scripts and other information on the Client computer.

Value: *path*

Default: %SSOINSTALLDIRE%\data

UserCacheDirectory

Defines the location of the cache directory that stores the user-specific logon credentials on the Client computer.

Value: *path*

Default: %SSOINSTALLDIRE%\data

ApplicationScriptCachePeriod

Defines how long the SSO Client stores SSO Scripts before refreshing them. You may want to use this reduce network traffic.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 0d

MessageOfTheDayCachePeriod

Defines how long the SSO Client will store SSO application messages of the day. You may want to use this to reduce network traffic.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 0d

ServerPublicKeyFile

Defines where the SSO Server's public key is stored.

Value: *path and file name*

Default: P%SSOINSTALLDIRE%\data\server_key.ini

CacheFileMaxAge

Defines the maximum age of the files in the cache directory. Any data file older than this will be deleted.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 30d

[EventCommands]

SsoSignOn

Defines the Windows command-line program or script to run when a user logs on to the SSO Client. In versions earlier than SSO r8.1 this token was called UserLogonCmd.

Value: *path and script or command name*

Default: [no default]

SsoSignOff

Defines the Windows command-line program or script to run when the user logs off (or is forced to log off) the SSO Client. In versions earlier than SSO r8.1 this token was called UserLogoffCmd.

Value: *path and script or command name*

Default: [no default]

WindowsLogon

Defines the Windows command-line program or script to run when a user logs on to the computer and authenticates using the SSO GINA or Windows GINA. This coincides with the Windows "Loading personal settings..." dialog when logging-in.

Value: *script name*

Default: [no default]

StartShell

Defines the Windows command-line program or script to run after the Windows "Applying personal settings..." dialog has disappeared, but before the user's shell has become visible.

Value: *path and script or command name*

Default: [no default]

WindowsLogoff

Defines the Windows command-line program or script to run when a user has selected logoff and after their Windows Shell has exited -- it cannot be used to clean-up applications. Its execution coincides with the Windows "Logging off..." dialog.

Value: *path and script or command name*

Default: [no default]

Lock

Defines the Windows command-line program or script to run when Windows is locked. This command is executed once the interactive desktop has changed to Winlogon but before the Ctrl-Alt-Del dialog has appeared.

Value: *path and script or command name*

Default: [no default]

Unlock

Defines the Windows command-line program or script to run after the user has been authenticated by the Windows GINA, but before the interactive desktop is switched from Winlogon to Default.

Value: *path and script name*

Default: [no default]

MaxWait

Defines the time in seconds the CA SSO Client waits for an event command to finish before executing. In versions earlier than SSO r8.1 this token was called EventTimeout. If MaxWait is set to 0, the system returns immediately. If MaxWait is left blank, the CA SSO client will default the value to 60 seconds.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 60s

[Logging]

ConfigFile

Defines the location of the SSO Client logging configuration file. If the value is left blank, logging is disabled.

Value: *logging.properties file and path*

Default: %SSOINSTALLDIRE%\cfg\logging.properties

[NetworkCommunication]

ConnectTimeout

Defines the time in seconds the Client will try to connect to the SSO Server before it gives up.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 120s

ReceiveBufferSize

Defines the size (in bytes) of the largest file that can be downloaded by the Client. You must specify a value higher than the largest SSO application script (in bytes). This value must correspond with the SendBuffSize which is set on the SSO Server.

To set the SendBuffSize on a SSO Server use the Policy Manager and navigate to Configuration Resources, SSO Server Settings, Communication, SendBuffSize and edit this value.

Value: *number of bytes*

Default: 262144

ListenPort

Defines on which port the Client listens for messages from the SSO Server. This affects Session Management. You may need to allow this port number on Client firewalls.

Value: *port number*

Default: 20001

Disconnect

Defines whether the Client disconnects from the SSO Server after every major operation. This frees up server resources and may improve performance in large installations.

Value: [yes|no]

Default: yes

IdentityFile

Defines the path to the certificate file (.pem file) which is used for TLS communication.

Value: *pathname*

[OfflineOperation]

Enabled

Defines whether you want to enable Offline Operation. This lets users connect to CA SSO when the SSO Client cannot establish connection to the SSO Sever and/or the authentication agent.

Value: [yes|no]

Default: yes

TimeLimit

Defines the maximum time that a user may continue to use CA SSO while offline before they must connect to the network and re-authenticate.

Value: *time in seconds (s), minutes (m), hours (h), or days (d)*

Default: 5d

RetryServer

Defines how frequently the SSO Client attempts to reconnect to the SSO Server if the SSO Client is offline. This should be a short enough time that the SSO Client can restore the connection in a timely manner, but long enough so it does not create too much network traffic.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 60s

[ScriptInterpreter]

Plugins

Defines which additional Tcl extensions (DLLs) the SSO Interpreter must load. The Plugins DLLs must live in the SSO Client directory, or in a directory listed in the PATH environment variable. The DLLs are separated by spaces.

Value: *DLL names*

Default: [no default]

VarsOverwrite

Defines the default values for some of the SSO Interpreter values. For example:

VarsOverwrite=_TIMEOUT=10 _WINTITLE=”Windowtitle”

The maximum number of name/value pairs that will be read when parsing is 32. The total string length for VarsOverwrite that will be read in is 254. For more details about the variables, see the *Tcl Scripting Reference Guide*.

Value: *variable name and value*

Default: [no default]

[HLLAPI]

Hllapi

Defines the name of the HLLAPI DLL provided with the emulation software (excluding the path).

For example: Whllapi.dll

Value: *DLL name*

Default: [no default]

HllapiFunc

Defines the name of the function that SSO calls to execute HLLAPI services.

Consult the emulation software documentation or vendor for the function name.

For example: WinHLLAPI

Value: *function name*

Default: [no default]

HllapiDllPath

Defines the location of the HLLAPIDLL folder (do not include the file name).

For example: "C:\Program Files\QWS370 PLUS"

Value: *path*

Default: [no default]

[WebAgentIntegration]

CookieURLs

Defines the fully-qualified URL of the site on which the web server and SSO web agent are running. For example: http://computer1.ca.com

Value: *HTTP URL*

Default: [no default]

SSOTokenVersion

Defines the SSO web token version supported on the web server. This is dependent on the version of Siteminder that you integrate with SSO Client. For Siteminder r12 and above, the SSO web token version must be 12. For Siteminder versions earlier than r12, the SSO web token must be 6. Cookies for target Siteminder versions are created based on this value.

Value: [6|12]

Default: 6

[ConfigurationSource]

ClientIniFile

Defines the location on the network of the central Client.ini file.

Value: *path and file name*

Default: [no default]

AuthIniFile

Defines the location on the network of the central Auth.ini file.

Value: *path and file name*

Default: [no default]

CachePeriod

Defines how frequently the SSO Client should download the central SSO Client configuration files (Client.ini and Auth.ini) from the central network location.

Value: *time in [days]d[hours]h[minutes]m[seconds]s*

Default: 1d

[CredentialProvider]

EnableSSOCredentialProvider

Defines how the Credential Providers will be loaded based on this key. If yes is specified, Credential Providers are loaded. If set to no, the Credential Providers are not loaded.

Value: *[yes/no]*

Default: yes

AddLocalMachineToDomainSelectionOptions

Defines how the local machine name will be added to the Domain List box for selection. If yes is specified, the local machine name is added to the Domain list box for selection. If set to no, only the domain will be listed in the list box.

Value: *[yes/no]*

Default: yes

FetchDomainsFromSystem

Defines from where the Domains are fetched. If yes is specified, the Domains are fetched from the network. If set to no, the Domains are fetched from the Key Domains INI file.

Value: *[yes\no]*

Default: yes

FilterMSProvider=

Default value is yes. If yes, the Microsoft Credential Provider is filtered and SSO Microsoft Credential provider wrapper is shown. If no, then the Microsoft Credential Provider if available for logon to the machine. This parameter is used only if EnableSSOCrdentialProvider parameter is set to Yes; else, this parameter is ignored.

Value: *[yes/no]*

Default: yes

Domains

This key is used only if the FetchDomainsFromSystem key is set to no.

Value: *list of domains separated by spaces.*

Default: [no default]

LoginBitmap

Defines the path (including the file name) of an image to use for the SSO credential provider logon window.

If this value is omitted, or that the image cannot be loaded, the credential provider uses a default bitmap embedded in the resource file.

Value: *path and image name*

Default: [no default]

[CredentialProviderTileImages]

ServerSetTile

Defines the path (including the file name) of an image to use for the server set tile. The image must be a .bmp file.

Value: path and image name

Default: [no default]

MSCPTile

Defines the path (including the file name) of an image for the Windows only logon dialog. If this value is omitted, or the image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key. If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded. The image must be a .bmp file.

Value: path and image name

Default: [no default]

SSOCPTile

Defines the path (including the file name) to use for the CA SSO credential provider authentication dialog. If this value is omitted, or the image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key. If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded. The image must be a .bmp file.

Value: path and image name

Default: [no default]

WINCPTile

Defines the path (including the file name) of an image to use for the Windows authentication dialog. If this value is omitted, or the image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key. If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded.The image must be a .bmp file.

Value: path and image name

Default: [no default]

LDAPCPTile

Defines the path (including the file name) of an image to use for the LDAP credential provider authentication dialog. If this value is omitted, or that image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key. If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded. The image must be a .bmp file.

Value: path and image name

Default: [no default]

RSACPTile

Defines the path (including the file name) of an image to use for the RSA credential provider authentication dialog. If this value is omitted, or that image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key.If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded. The image must be a .bmp file.

Value: path and image name

Default: [no default]

CERTCPTile

Defines the path (including the file name) of an image to use for the Certificate credential provider authentication dialog. If this value is omitted, or that image cannot be loaded, the credential provider uses the bitmap specified in the LoginBitmap key.If LoginBitmap key is also empty or cannot be loaded, a default image embedded in the resource file is loaded. The image must be a .bmp file.

Value: path and image name

Default: [no default]

Registry Configuration

Use the following registry file to define settings that affect the user experience.

Client registry Key

HKLM\Sofware\ComputerAssociated\SingleSignOn\Client

CommMode

Specifies a dword value with takes 0, 1 or 2

- 0 - Compatible Mode
- 1 - TLS Communication Mode
- 2 - FIPS compatible TLS Communication Mode

After the SSO Client has been installed, if the communication mode needs to be changed, this registry key must be changed accordingly, and the CA SSO Client service restarted. Also, the path for the IdentityFile key in the client.ini and the chosen authentication agents must be set in case of mode 1 and 2 mentioned above.

Shared Workstations

You can configure the CA SSO Client to suit how people use their computers. For example, you might have a kiosk-style workstation environment where lots of people access a single computer each day. Or you might have one computer per person.

These computer modes affect how the computer is unlocked and how the users of that machine access their SSO-enabled applications and the Windows desktop. You must make sure you understand how users access computers in your organization before you decide which mode to work with.

The computer modes are:

Single-user workstation mode

This is used in non-shared workstation environment. The computer can only be unlocked by the person who locked it (or a systems administrator). This therefore suits a situation where the same person uses this computer all the time. This option provides the greatest security.

Scenario

Nancy sits at one workstation full-time. She does not share her workstation with anyone else. She is the only person who logs into the domain and uses CA SSO from this computer.

Fully Shared Workstation/Mode 3 (kiosk)

This is used in a full shared computer environment. The computer can be unlocked by any SSO user, but there is no reference to the underlying Windows user. This suits a situation where two or more people share a computer and Windows setup, but each user wants to have their own customized CA SSO applications. This is like the semi-shared workstation mode 1 option, but is much faster and suits an environment where several people may have to use one computer in quick succession.

In Vista, you can configure the credential provider to log in to the Windows desktop automatically upon startup using the Windows username and password stored in the registry keys for Credential Provider. So, users share the same underlying Windows credentials but different CA SSO credentials.

Scenario

A busy ward in a hospital has a computer and printer for general use. This computer is used by many doctors who need to access data quickly and print things out without going back to their offices. This computer has a generic Windows desktop and settings that connect to the printer. Each doctor can unlock the computer and see their CA SSO applications. From here they can print to the printer right beside the computer because this Lock Option uses the local settings of the shared Windows setup.

Semi-shared workstation/Mode 1 (Single Windows desktop for all users)

This mode caters to multiple users on one computer who can share a single Windows desktop. This is used in a semi-shared computer environment. The computer can be unlocked by any SSO user, as long as that user shares the underlying Windows logon. This suits a situation where two or more people share a customized Windows setup, but need access to their own CA SSO applications on a workstation. All users work as different CA SSO users using the same Windows profile.

You must write a logoff script for each user. When either user is logged off, their logoff script runs, closing all of their open applications.

Scenario

Hillary and Mike both work in Human Resources and spend a lot of their time in interviews, so they share one workstation. They share a Windows desktop that shows the applications that relate to their job, but they need to have separate access to their own CA SSO applications. When either of then unlocks the workstation in their own name they can see the same Windows setup, but their own specific CA SSO applications.

Semi-shared workstation/Mode 2 (Unique Windows desktops for each user)

This mode caters to multiple users on one computer who all need their own Windows Desktop.

This is used in a semi-shared computer environment. The computer can be unlocked by any SSO user, but if the new user has a different underlying Windows logon, the old Windows user is logged out and the new user is logged on. This suits a situation where two or more people share a computer and each user wants to have their own customized Windows setup as well as their own CA SSO applications. This method is slower, because it completely logs one user off Windows and then logs the next user on and is not recommended.

You must write a logoff script for each user. When either user is logged off, their logoff script runs, closing their open applications.

Scenario

Peter and Sally both share a workstation. They do very different jobs so they each want their own Windows desktop and their own CA SSO sessions. Peter works in the morning and leaves at midday. When Sally starts work in the afternoon she unlocks the workstation in her own name and sees her own Windows desktop and her own CA SSO applications.

Single user Windows session/Mode 4 (Single CA SSO user per Windows session)

This mode is used in a non-shared Windows session environment (one CA SSO user for one Windows user, but multiple Windows sessions (accounts) are allowed).

A Windows session corresponds to exactly one Windows account. The Windows account is not shared by multiple SSO users. The locked Windows session can only be unlocked with the credentials that match those of the last SSO user. If the SSO user that unlocks the station has different Windows credentials, a new Windows session is opened without logging the previous Windows user off. If this session is locked too, each of the signed-on SSO users can only unlock the Windows session associated with it. Other SSO users can log in, only if they have different Windows users. (A new Windows session is created.) This means that multiple users share the same machine resources.

Scenario

Sharon logged into her Windows session to update a patient's file and then she locked her Windows session. George, who has different underlying Windows credentials logged onto the machine, using his credentials, which gave him access to his new Windows session and has now locked his Windows session. Both users can interchangeably use the machine just by unlocking their corresponding sessions without seeing each others applications.

Multiple users Windows session/Mode 5 (Multiple SSO users per Windows session)

This mode is used when two or more people share the same Windows account but need access to their own SSO applications on a computer (multiple SSO users for one Windows user, but multiple sessions are allowed).

The locked Windows session can be unlocked by any SSO user as long as that user has the same underlying Windows user. If the unlocking SSO user has a different underlying Windows user, a new Windows session is opened without logging off the previous Windows user. This means that multiple users share the same machine resources.

Scenario

Jack and Roger share the same Windows account, but each needs to access their own SSO applications on a computer. Initially, Jack logs into the Windows account using SSO Credential Provider and locks the Windows session. Roger chooses to login using SSO Credential Provider and logs in to the same Windows session that Jack was logged in to. By this time, Jack is logged off from SSO and Roger logs in. Roger cannot see any of Jack's applications because the logoff scripts configured for Jack run immediately after Jack logs off SSO, thereby closing all of Jack's applications. If Jarrod logs on using different Windows credentials, a new Windows session is opened.

Note: Modes 4 and 5 are only supported on Microsoft Vista platforms with SSO Credential Provider installed.

Configure Shared Computer Mode

This procedure explains how to configure all six shared workstation modes. To activate shared workstation modes, you must create a shared user account and configure the Windows "AutoAdminLogon" setting in the Windows registry.

To configure shared workstation mode

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [GINA/StationLock] section.

3. Edit the following values:

   UnlockStationMode

   Defines the workstation mode. These values only apply when the SSO GINA and Credential Providers are in use. When GINA is in use, the valid values are zero through three. For Credential Providers the the valid values are zero through five.

   - Single user lock option

     Select option 0. This is used in a regular non-shared computer environment (one Windows user, one SSO user).

   - Multiple SSO user lock option
     Select option 1. This is used when two or more people share a customized Windows setup, but need access to their own SSO applications on a computer (one Windows user, multiple SSO users).

   - Multiple Windows user lock option
     Select option 2. This is used when more than one person shares a computer and each user needs to have their customized Windows setup as well as their own SSO applications (multiple Windows users, multiple SSO users).

   - Kiosk mode lock option
     Select option 3. This is used when more than one person shares a computer and shares a generic Windows setup, but each user needs to have their customized SSO applications. This is like option Multiple Windows user lock mode, but is much faster and suits an environment where several people may have to use one computer in quick succession or may not have a Windows user account.

   - Single user Windows session lock option
     Select option 4. This is used in a non-shared Windows session environment (one CA SSO user for one Windows user, but multiple Windows sessions (accounts) are allowed)

- Multiple users Windows session lock option
  Select option 5. This is used when two or more people share the same Windows account but need access to their own SSO applications on a computer (multiple SSO users for one Windows users, but multiple sessions are allowed).

  Value: [0|1|2|3|4|5]

  Default: 1

4. Save the Client.ini file.

Note: If you use this Lock Option, you must set the "GINAPassThrough" parameter to ‘yes’ in the Client.ini file.

You must also configure the Windows Registry to automatically log the defined Windows user onto the computer. To do this, edit the "AutoAdminLogon" setting in the Windows registry. You must turn this on so that the user is never prompted to enter the Windows password.

Hints and Tips for Shared Computer Mode

Locking the computer on startup

After auto-logging into the computer, you may want to invoke the CA SSO Client and Station Lock automatically. This means that an end user has to pass SSO primary authentication before obtaining access to the desktop.

You can lock the computer using several methods - two options are listed below.

- Define a shortcut in the \Documents and Settings\All Users\Start Menu\Programs\Startup folder to invoke the CA SSO Client interface
- Add String value entries to the HKLM\Software\Microsoft\Windows\Current Version\Run registry key to invoke the CA SSO Client

Hiding the desktop when a user is logging out of CA SSO

In a multi-user environment, ensure that one user does not view another user’s SSO applications. This is possible in a Shared Computer environment when one user logs onto CA SSO from Station Lock when a previous user was already logged on. CA SSO invokes the first user’s logoff script which may take a few seconds to run; during this time the second user can see the data that user had previously displayed on screen.

CA SSO automatically comes with a utility that you can use to hide the desktop while a logoff script is running on the user's computer.   If you invoke this utility at the very start of the logoff script, it "covers" the screen and all open windows with a selected color and an information message, or an image file. You must also invoke the utility at the very end of the logoff script (or whenever the logoff script is to terminate) to remove the "cover".

The utility is called hidedesktop.exe and you can find it in the CA SSO Client installation directory. See the hidedesktop.html help file in the same directory for an overview on how to use it.

Note: This utility is designed to improve end user experience, but it may not work in all circumstances. For example, if one of the open programs is defined to run in a "stay-on-top" window, then the hidedesktop.exe may not be able to "cover" that application window.

How to Customize Log In Behavior

In shared computer modes you are able to customize the way you log into CA SSO by configuring different settings to display only the screens you want to access.

To configure the log in process for a user, do the following:

- Customize the Default Login screen.
- Customize Workstation lock and unlock behavior.
- Configure display of CTRL+ALT+DEL screen.
- (Optional) Disable Fast User Switching.
- (Optional) Configure system to login Windows users automatically during system start up.

Customize Default Login Screen

Using CA SSO you can customize the default login screen that is displayed to users. The login screen can be customized to reflect a custom image, custom text, default authentication method, and default domain. The Client.ini and Auth.ini files control the behavior of the default login screen.

The custom images to be displayed on the login screens are controlled by the [GINA] and [CredentialProviderTileImages] sections of the Client.ini file.

To customize the default login screen displayed to users

1. Open the Auth.ini file and configure the following parameter:

   - ServerSetSelection

     If set to 0 or 1, the Server Set dialog is displayed and you can change the authentication method and domain used for authentication.

     If set to 2 or 3, the Server Set dialog is not displayed, the user is taken directly to the login screen and the login screen uses the default authentication method, serverset,   and domain.

     Note: If a different non-default auth method, server set or domain needs to be used during initial login use "Change Credentials" button to get to the Server Set Selection Dialog

2. Configure the following parameters in the Client.ini file to set the default domain that is used for logging users:

   - FetchDomainsFromSystem
   - Domains

   Note: For more information on the ServerSetSelection parameter, see the Auth.ini Configuration (see page 153) section.

More Information:

Auth.ini Configuration (see page 153)
Client.ini Configuration (see page 163)

Customize Workstation Lock and Unlock Behavior

Using CA SSO you can customize the default lock and unlock behavior of the login process.

To customize the lock and unlock behavior, open the Client.ini file and configure the the UnlockStationMode parameter:

- For GINA, use values 0-3.
- For Credential Providers, use values 0-5.

For more information on the lock and unlock differences in shared workstation modes 0-5, see Shared Workstations (see page 31).

Configure Display of CTRL+ALT+DEL Screen

During the login process, you can configure Windows to hide the CTRL+ALT+DEL screen for users while displaying a default login window. The following steps describe how to replace the traditional CTRL+ALT+DEL screen that Windows displays on a system reboot or lock with your customized CA SSO authentication tile:

To disable CTRL+ALT+DEL screen

1. Set the value of the following parameter to 0 (zero) in the Client.ini file.

   LogonCAD

2. Use Regedit to edit the following registry key:

   HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

3. Create the following DWORD in the previous key and set its value to 1:

   DisableCAD

   The CTRL+ALT+DEL screen is suppressed during system reboot or system lock.

Disable Fast User Switching for Shared Modes 0 through 3

Fast user switching is a feature in Windows XP and Windows Vista that lets you switch between users sharing the same computer without logging off any user. Multiple users can share the same computer without closing the programs of other users.

In Windows XP, fast user switching is supported, but is not available on machines that are members in Windows domains.

In Vista, fast user switching is enabled by default.

Fast user switching is not very useful if you are using modes 0-3 on Vista. The number of user sessions per Windows user is restricted to one in these modes.

To disable fast user switching

1. Use Regedit to edit the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
   ```

2. Create the following DWORD in the previous key:

   ```
   HideFastUserSwitching = [0|1]
   ```

   where

   0

   Fast user switching is enabled.

   1

   Fast user switching is disabled.

   The Switch User button is suppressed on logon and unlock screens.

Disable Fast User Switching Functionality using a Group Policy

You can disable fast user switching functionality in Windows Vista by editing the associated group policy.

Note: The Group Policy Editor does not exist in certain editions of Windows Vista.

To disable fast user switching

1. Click Start, Run, enter the following command, and click OK.

   ```
   gpedit.msc
   ```

   The Group Policy Editor appears.

2. Click Local Computer Policy, Administrative Templates, System, Policy, and enable the following parameter:

   ```
   Hide entry points for Fast User Switching
   ```

3. Exit the Group Policy Editor.

   The fast user switching functionality is disabled.

Customize Auto Login of Users During System Startup

You can configure the credential provider to log in to the Windows desktop automatically upon startup using the Windows username and password stored in the registry keys. The behavior of automatic login is controlled by the following registry key hive:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.
```

Configuring Log in Process for a Shared Workstation Mode 3 in a Hospital Example

A busy ward in a hospital has a computer and printer for general use. This computer is used by many doctors who need to access data quickly and print things out without going back to their offices. This computer has a generic Windows desktop and settings that connect to the printer. Each doctor can unlock the computer by entering their username and password only and see their CA SSO applications. From here they can print to the printer right beside the computer because this Lock Option uses the local settings of the shared Windows setup. If a doctor locks the computer and other doctors wants to log in to the computer, the other doctors must be able to unlock the computer just by entering their respective username and password.

Note: The following procedure is valid on Windows Vista.

To configure the login process for workstation mode 3:

1. (Optional) Set the unlock workstation mode as follows:

   ```
   UnlockStationMode = 3
   ```

2. Customize the login screen to never display the server set as follows:

   ```
   ServerSetSelection = 3
   ```

   If Server Set Selection screen is disabled CA SSO automatically populates the default server set, authentication method, and domain and displays only the Username and Password fields to the user on the login screen.

3. (Optional) Disable CTRL+ALT DEL

   Note: When you disable CTRL+ALT+DEL screen, the login screen that you have customized for your users appears.

4. (Optional) Disable Fast User Switching entry points.

Note: This example uses shared workstation mode 3 which means that all SSO users share the same Windows profile. An administrator needs to write a logoff script that runs and closes all open applications when a user logs off.

More Information

How to Customize Log In Behavior (see page 194)

Offline Operation

This section helps you enable and configure offline operation for the CA SSO Client.

Enable/Disable Offline Operation

Offline operation lets users log onto CA SSO and launch SSO-enabled applications when the SSO Client cannot connect to the SSO Server.

To enable/disable offline operation

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [OfflineOperation] section.

3. Edit the following value:

   Enabled

   Defines whether you want to enable Offline Operation. This lets users connect to CA SSO when the SSO Client cannot establish connection to the SSO Sever and/or the authentication agent.

   Value: [yes|no]

   Default: yes

4. Save the Client.ini file.

   Note: If modified on a local machine, we recommend that you shut down the CA SSO Client (including SSO Tools, SSO Status Icon and SSO Launchbar) before you make any changes to the configuration files. If you do not shut down the CA SSO Client, you must restart it for changes to take effect.

Mark an Application for Offline Use

You can mark any SSO-enabled application for offline use. This means that the user can log on to this application while the CA SSO Client is unable to connect to the CA SSO Server or authentication agent.

To mark an application for offline use

1. Open the Policy Manager.
2. Navigate to Application Resources, Application.
3. Double-click the application you want to mark for offline use.
4. Select the Attributes icon.
5. Select the Available Offline check box.

Change the Offline Cache Period

To enable offline operation, the CA SSO Client stores encrypted information in a local cache. You can change the expiry time of this cache. When this expires the user can no longer use offline functionality. You may want to decrease the expiration time to force users to log on to the server to re-verify their credentials more frequently. Conversely you may want to increase the time to let users access their applications offline if they are likely to be away from the network for an extended period of time.

To change the offline operation time

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [OfflineOperation] section.

3. Edit the following value:

   TimeLimit

   Defines the maximum time that a user may continue to use CA SSO while offline before they must connect to the network and re-authenticate.

   Value: *time in seconds (s), minutes (m), hours (h), or days (d)*

   Default: 5d

4. Save the Client.ini file.

Interface Configuration

You can configure the appearance and behavior of any of the three CA SSO Client interfaces (Launchbar, Status Icon, SSO Tools) using the Client.ini file.

Configure the Launchbar

You can change a number of options on the Launchbar interface.

To configure the Launchbar

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [Launchbar], [Launchbar/OptionsMenu], and [Launchbar/AppMenu] sections and edit the relevant values.

3. Save the Client.ini file.

Allow User Control over the Launchbar

You can let the user have some control over the appearance and behavior of the CA SSO Client Launchbar. You might want to limit this functionality if you have a shared computer environment and you want a standard configuration. Administrators can control:

- Whether the user sees the Options button
- What options the user can access when they select the Options button. The available options are:
  - Exit button
  - Application Refresh button
  - My details button, which lets the user change their primary authentication password (if this is supported by that authentication method)
- Whether the Launchbar is always on top of other applications
- Whether the Launchbar will automatically hide when the mouse is not over it

To configure what control the user has over the Launchbar

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [Launchbar] section.

3. Edit the following value:

   DisplayOptionsButton

   Defines whether the user sees an Options button on the Launchbar. When the user clicks the Options button they will see options to:

   - Exit the Launchbar
   - Refresh their application list
   - Display their details
   - Auto hide the LaunchBar, if docked
   - Display the LaunchBar on top of other windows

   Value: [yes|no]

   Default: yes

4. Find the [LaunchBar/OptionsMenu] section.

5. Edit the following values:

   AlwaysOnTop

   Defines whether the "Always on top" item appears in the Options menu on the Launchbar. This lets the user control the behavior of the SSO Launchbar and overrides the AlwaysOnTop token in the Launchbar section.

   Value: [yes|no]

   Default: yes

   AutoHide

   Defines whether the "Auto Hide" item appears in the Options menu on the Launchbar. This lets the user control the behavior of the SSO Launchbar and overwrites the AutoHide token in the Launchbar section.

   Value: [yes|no]

   Default: yes

   RefreshApplist

   Defines whether the "Refresh Application List" item appears in the Options menu on the Launchbar. This lets the user refresh their application list. You might set this value to no if you wanted to reduce the load on the servers in a large-scale implementation.

   Value: [yes|no]

   Default: yes

   UserConfiguration

   Defines whether the "My Details" item appears in the Options menu on the Launchbar. This lets the user change their authentication credentials, if the authentication method supports this.

   Value: [yes|no]

   Default: yes

   Exit

   Defines whether the Exit Launchbar item appears in the Options menu on the Launchbar. This lets the user exit from CA SSO.

   Value: [yes|no]

   Default: yes

6. Save the Client.ini file.

Change the Launchbar Position and Size

You can define where you want the Launchbar window to appear when it is launched. You will probably base this decision on what you think is most convenient for your users.

To change the Launchbar Position and Size

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [Launchbar] section.

3. Edit the following values:

   StartDocked

   Defines whether the Launchbar starts docked to one edge of the screen, or whether it is free-floating. The dock location is defined by the DockedEdge token.

   Value: [yes|no]

   Default: no

   DockedEdge

   Defines which edge of the screen the Launchbar will start docked to if you specified StartDocked=yes.

   Value: [top|bottom|left|right]

   Default: top

   OffSetX

   Defines the X coordinate (horizontal plane) of the Launchbar from the top left corner of the dialog to the top left corner of the screen. This value only applies to the SSO Launchbar if it is in "floating" mode. If StartDocked=yes, this value is ignored.

   Value: *The number of pixels*

   Default: the window is centred

   OffSetY

   Defines the Y coordinate (vertical plane) of Launchbar from the top left corner of the dialog to the top left corner of the screen. This value only applies to the SSO Launchbar if it is in "floating" mode. If StartDocked=yes, this value is ignored.

   Value: *number of pixels*

   Default: the window is centred

AlwaysOnTop

Defines whether the Launchbar should always be visible and stay on top of all other windows. This value is only valid if StartDocked=no.

Value: [yes|no]

Default: no

AutoHide

Defines whether the Launchbar should automatically hide until the mouse moves over it. This token is only valid when the Launchbar=yes.

Value: [yes|no]

Default: no

4. Save the Client.ini file.

Configure the SSO Status Icon

You can change a number of options for the SSO Status Icon that appears in the user's Windows tool bar.

To configure the SSO Status Icon

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [StatusIcon] and the [StatusIcon/Menu] sections and edit the relevant values.
3. Save the Client.ini file.

Configure SSO Tools

You can change a number of options on the CA SSO Client SSO Tools interface.

To configure SSO Tools

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [SSOTools] section and edit the relevant values.
3. Save the Client.ini file.

Configure SSO GINA

Users can log onto Windows locally or through a domain (network) using the SSO GINA.

For users to log onto a network (Windows domain), you must create a domain application on the CA SSO Server, for example, MYCORPDOMAIN (where MYCORPDOMAIN is the name of your domain) and assign users to the application. For users to log onto Windows locally, they can use the default NT_LOCAL_LOGON application, or alternatively, create a local logon application, for example MYLOCALLOGON (where MYLOCALLOGON is the hostname of your machine) and assign users to the application.

When an SSO user logs on to the network (Windows domain), the SSO GINA looks for the domain application, for example, "MYCORPDOMAIN". If the SSO GINA cannot find the application, it uses the NT_LOCAL_LOGON application to log users onto Windows locally. Similarly, when a user wants to log onto Windows locally, the SSO GINA looks for the local logon application if it exists, for example MYLOCALLOGON. If it cannot find the application, it uses the default NT_LOCAL_LOGON application.

Note: The above example assumes the use of MYCORPDOMAIN and NT_LOCAL_LOGON/MYLOCALLOGON for those intending to use SSO for network (Domain) and local logon respectively.

To set up an application for the SSO GINA on the CA SSO Server

1. Open the Policy Manager
2. Create an application and name it according to your naming standards, for example MYCORPDOMAIN.

   Note: For security reasons, we strongly recommend that you make the domain application a sensitive application. This forces users to re-authenticate before changing their domain application password. In this case the sensitive timeout is never set to zero (by default this is set to five).

3. Assign this application to all user(s).

More information:

Configuring the CA SSO Client (see page 151)
Auth.ini Configuration (see page 153)
Client.ini Configuration (see page 163)
Registry Configuration (see page 188)

Change the SSO Windows Authentication screens (GINA)

You can change the appearance of the SSO GINA to control the user experience. You might choose to alter this in a shared computer environment.

To change the appearance of the Windows logon screens using the SSO GINA

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [GINA] section.

3. Edit the following values:

   LogonBitmap

   Defines the name (including the path) of an image to use for the SSO GINA's logon window.

   If this value is omitted, or that image cannot be loaded, the GINA uses a default bitmap which says “Welcome to CA Single Sign-On". You can customize this text using LogonText value.

   Value: *path and image name*

   Default: [no default]

   LogonTitle

   Defines the title for the SSO GINA's logon window. If not specified, the default value is “Windows Logon”.

   Value: *dialog title*

   Default: Windows Logon

   LogonText

   Defines the text the user will see in the SSO GINA Logon dialog.

   Value: *Text*

   Default: Welcome to CA Single Sign-On

   Font

   Defines the font used for LogonText and LockedText on the GINA windows.

   Value: *any system font by name*

   Default: Arial

FontSize

Defines the size of the font specified in the Font value.

Value: *font size*

Default: 13

GinaPassThrough

Defines whether to bypass the SSO GINA and go to the Microsoft GINA (MSGINA), even if the SSO GINA is installed.

If GinaPassThrough is set to no, the user sees the SSO GINA in all cases (welcome screen, logon screen, Ctrl+Alt+Del options screen and locked screen).

If GinsPassThrough is set to yes, the user sees the MSGINA screen for the logon screen, but sees the SSO GINA for all other GINA screens.

You need to use this setting if you plan to use Full Shared Computer mode. This is used when more than one person shares a computer and each user can share a generic Windows setup, but each user needs to have their own customized SSO applications.

Value: [yes|no]

Default: no

LogonCAD

Specify whether to display the Ctrl + Alt + Del dialog at logon.

Value: [yes|no]

Default: yes

FetchDomainsFromSystem

Defines from where the Domains are fetched. If yes is specified, the Domains are fetched from the network. If set to no, the Domains are fetched from the Key Domains INI file.

Value: [yes|no]

Default: yes

Domains

This key is used only if the FetchDomainsFromSystem key is set to no.

Value: *list of domains separated by spaces.*

Default: [no default]

4. Save the Client.ini file.

Change the Appearance of the SSO Windows Locked Screen (GINA)

You can change the appearance of the SSO GINA to control the user experience. You might choose to alter this in a shared computer environment.

To change the appearance of the Windows locked screens using the SSO GINA

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [GINA] section.

3. Edit the following values:

   LockedBitmap

   Defines the name (including the path) of an image to use for the SSO GINA's 'Station locked' window.

   If this value is omitted or that image cannot be loaded, the GINA uses a default SSO bitmap which says "This computer is in use and has been locked".

   Value: *path and image*

   Default: [no default]

   LockedTitle

   Defines the title for the SSO GINA's 'Station locked' window. If not specified the default value of "Computer Locked" is used.

   Value: *dialog title*

   Default: Computer Locked

LockedText

Defines the text that the user sees on the SSO GINA when the computer is locked.

Value: *text*

Default: This computer is in use and has been locked.

Font

Defines the font used for LogonText and LockedText on the GINA windows.

Value: *any system font by name*

Default: Arial

FontSize

Defines the size of the font specified in the Font value.

Value: *font size*

Default: 13

4. Save the Client.ini file.

Windows Registry Values

The following registry setting determines whether the CTRL+ALT+DEL screen is displayed within Windows:

HKEY_LOCAL_MACHINE\SOFTWARE\>Microsoft\Windows NT\WinLogon\DisableCAD

If logonCAD is set to 0 or 1, CA SSO overrides this value. If logonCAD is set to 2, then whatever value the DisableCAD registry setting is determines if the CTRL+ALT+DEL screen is displayed).

The following registry setting determines the action that the SSO GINA takes when you remove the smart card from the reader after you have logged on with certificate authentication method and a smart card.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\WinLogon\scremoveoption

If scremoveoption is set to 0, the GINA Security dialog is displayed. If it is set to 1, the workstation is locked. If it is set to 2, a Windows logoff is performed.

When you install SSO GINA, a registry key--ginadll is added to the following registry hive:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\WinLogon

ginadll

Specifies the path of the GINA DLL that is used to log a user into the computer. This key is of type REG_SZ.

Important! The ginadll value must point to a valid GINA DLL. If the path points to an invalid GINA DLL, users might not be able to log into the computer.

Configure SSO Credential Provider

This agent is available only for Microsoft Windows Vista. Users can log onto Windows Vista locally or through a domain (network) using the SSO Credential Provider.

Creating the domain and local login applications for SSO Credential Providier is very similar to how you create them for GINA.

To create a domain or local login application, and to set up an application for the SSO Credential Provider on the CA SSO Server, see Configure SSO GINA (see page 206) for more information.

More Information

Configuring the CA SSO Client (see page 151)
Auth.ini Configuration (see page 153)
Client.ini Configuration (see page 163)
Registry Configuration (see page 188)

Configure SSO Tools

The SSO Credential Provider can be enabled or disabled once it has been installed. Also, when using SSO, it is possible to filter out all other Credential Provider tiles during Windows logon, thus enabling only the SSO Credential Provider tile.

To configure these options

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [CredentialProvider] section and edit the relevant values.
3. Save the Client.ini file.
4. Reboot the machine after changing these settings (Recommended).

Windows Registry Values

When you install SSO client with credential providers the following keys are added to the registry hive,

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication

Credential Providers

Specifies the keys of all the credential providers installed on your computer. The credential provider keys are of REG_SZ type. The Class ID(CLSID) of the Credential Provider is as follows:

{781A7B48-79A7-4FCF-92CC-A6977171F1A8}

Credential Provider Filters

Specifies the CLSID of the credential provider filters installed on your computer. The CLSID of the credential provider filter is as follows:

{A54AB0C2-B99A-43ff-A897-865D141960B6}

MSCPWrapper

The CLSID of MSCPWrapper is as follows:

{7BBBD3CE-A872-4c35-BCDB-16A78828703E

The following registry keys are added to the registry hive,

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

DefaultUserName

Specifies the underlying Windows username for a computer shared in a Kiosk mode. By default, the value for this key is empty. When a CA SSO user logs into the computer for the first-time in shared workstation mode 3, the underlying Windows username is captured and stored as the value for this registry key.

DefaultDomainName

Specifies the underlying Windows domain name for a computer shared in a Kiosk mode. By default, the value for this key is empty. When a CA SSO user logs into the computer for the first-time shared workstation mode 3, the domain name of the underlying Windows user is captured and stored as the value for this registry key.

DefaultPassword

Specifies the password for the underlying Windows username of an CA SSO user who is using the computer in a kiosk mode. By default, the value for this key is empty. When a CA SSO user logs into the computer for the first-time shared workstation mode 3, the underlying Windows password is encrypted using AES-128 bit algorithm and stored as the value for this registry key.

Note: If the registry keys DefaultUserName, DefaultDomainName, and DefaultPassword are empty, the user is prompted to enter any valid Windows user credentials. This Windows user credentials are then stored in the registry key.

SSOAutoAdminLogon

Specifies if the credential provider is configured to log in to the Windows desktop automatically upon startup using the Windows username and password stored in the registry keys. This key can take the following values:

0

Indicates that the credential provider does not automatically log into the Windows desktop upon startup.

1

Indicates that the credential provider uses the Windows username and password stored in the DefaultUserName and DefaultPassword to log into the Windows desktop on startup.

Default: 0

Application List Refresh Options

You can configure a regular application list refresh which you may choose to run at low traffic times. You can also enable an application list Refresh button that users can click at any time to refresh their applications.

Configure Automatic Application List Refresh

You can configure how often the CA SSO Client downloads the users' precalculated application lists from the CA SSO Server. To regularly calculate users' application lists on the CA SSO Server we recommend that you periodically run the psbgc utility using a scheduler.

To configure an automatic regular application list refresh

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [AppListRefresh] section.

3. Edit the following values:

   Enabled

   Defines whether you want to turn the SSO Client's automatic application list refresh on.

   If this is set to 'no', the rest of the tokens in this section are ignored.

   Value: [yes|no]

   Default: no

   Interval

   Defines the time between checks for an updated application list.

   Note: If this value is set, then the EarliestStartTime and LatestStartTime values are ignored.

   Value: *time in [days]d[hours]h[minutes]m[seconds]s*

   Default: [no default]

EarliestStartTime

In conjunction with LatestStartTime, defines the time period within which a daily refresh occurs (random point between EarliestStartTime and LatestStartTime). You might want to schedule this during low-network traffic periods.

If these values are set, they are only used if the 'Interval' token is not set.

The time is specified as a 24 hour clock: for example, 21:31 indicates 9:31 pm.

Value: *time in [hours]:[minutes]*

Default: 09:00

LatestStartTime

For a full description, see EarliestStartTime in this section.

Value: *Time in [hours]:[minutes]*

Default: 17:00

4. Save the Client.ini file.

Enable the Application List Refresh Button for Users

As well as configuring a regular application list refresh, you can also enable a button on the CA SSO Client interfaces that lets users trigger an application list refresh. You might choose to enable this option if you think users' application lists will change frequently and you want to let users update their application list in between scheduled updates. You might choose to disable this option if you have a large number of users and feel that they might overuse this button, which would generate large amounts of unnecessary network traffic and CA SSO Server processing.

You can enable the Application List Refresh button on any or all of the CA SSO Client interfaces. By default the Application Refresh option is enabled for all CA SSO Client interfaces.

To enable the Application List Refresh button on the CA SSO Client interfaces

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [LaunchBar/OptionsMenu] section.

3. Edit the following value:

   RefreshApplist

   Defines whether the "Refresh Application List" item appears in the Options menu on the Launchbar. This lets the user refresh their application list. You might set this value to no if you wanted to reduce the load on the servers in a large-scale implementation.

   Value: [yes|no]

   Default: yes

4. Find the [SSOTools] section.

5. Edit the following value:

   EnableRefreshButton

   Defines whether the Refresh List button is enabled on the SSO Tools interface. This button lets users update their application list from the SSO Server. You might set this value to 'no' if you wanted to reduce the load on the servers in a large-scale implementation.

   Value: [yes|no]

   Default: yes

6. Find the [StatusIcon/Menu] section.

7. Edit the following value:

   RefreshApplicationList

   Define whether to include the Refresh Application List option in the Status Icon menu. When a user clicks this the SSO Server recalculates that user's application list and sends it to the SSO Client. You might set this value to ‘no’ if you wanted to reduce the load on the servers in a large-scale implementation.

   Value: [yes|no]

   Default: yes

8. Save the Client.ini file.

Script Caching to Reduce Network Traffic

You can reduce network traffic by storing logon scripts in a cache on the CA SSO Client computer. If you enable script caching, each time a user launches an SSO-enabled application the logon script for that application is then stored on the CA SSO Client computer for a set period of time, for example a period of days. Within that time any user on that computer who launches that application invokes the cached logon script instead of contacting the CA SSO Server and downloading the logon script each time.

This functionality is separate from offline operation. Any application marked for offline operation automatically has its logon script cached, regardless of whether you enable script caching.

Note: Script caching does not store any private information such as logon credentials.

More information:

About the CA SSO Client (see page 125)

Enable SSO Script Caching

You can configure the CA SSO Client to cache SSO scripts, which reduces network traffic. If you enable script caching, then every time a user launches a particular application the CA SSO Client stores the SSO scripts for that application for a period of time. Whenever any user launches the application within the expiration period, the CA SSO Client uses the local script and the only load on the network is to retrieve the user's logon variables. For example, you might choose to set this value to expire after seven days.

This only applies to applications that are not marked for offline operation. For applications that are marked for offline operation, [OfflineOperation]\TimeLimit setting always overwrites the value set here.

To enable SSO script caching

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [Cache] section.

3. Edit the following value:

   ApplicationScriptCachePeriod

   Defines how long the SSO Client stores SSO Scripts before refreshing them. You may want to use this reduce network traffic.

   Value: *time in [days]d[hours]h[minutes]m[seconds]s*

   Default: 0d

4. Save the Client.ini file.

Change Retry Server Frequency During Offline Operation

Sometimes the connection between the SSO Client and the SSO Server may be down. When this happens the SSO Client periodically tries to reestablish connection. You can change how often the SSO Client tries to connect to the SSO Server.

To change retry server frequency

1. Open the Client.ini file.

   By default this is installed in the following location:

   C:\Program Files\CA\Single Sign-On\Client\cfg

2. Find the [OfflineOperation] section.

3. Edit the following value:

   RetryServer

   Defines how frequently the SSO Client attempts to reconnect to the SSO Server if the SSO Client is offline. This should be a short enough time that the SSO Client can restore the connection in a timely manner, but long enough so it does not create too much network traffic.

   Value: *time in [days]d[hours]h[minutes]m[seconds]s*

   Default: 60s

4. Save the Client.ini file.

Chapter 8: Configuring User Data Stores

CA SSO improves management of user passwords and access to applications; it is not a user management system in itself. It integrates with any third-party LDAP data store such as Microsoft Active Directory. CA SSO is designed to plug into your existing user management directory: you should continue to create and delete all users and user groups within your corporate directory. You should only use CA SSO to manage SSO-related information for those users and user groups, such as their authentication methods, application permissions, and session and password policies.

You can, however, create users and user groups in the CA SSO native User Data Store using the Policy Manager. You might choose to create individual users and user groups within CA SSO when you first install it so you can test basic functionality, but when you implement CA SSO in a live environment we expect that you will use your own user management system.

CA SSO embeds CA Directory, which is where it stores CA SSO-related user information such as the user's application passwords. If you choose, you can also use this instance of CA Directory as your primary user data store

This section contains the following topics:

Types of Data Store

There are two types of data store within CA SSO.

User Data Store

> This information is stored in CA Directory. If you already use a third-party LDAP data store such as Active Directory, CA Directory only stores CA SSO-specific information about users and links back to Microsoft Active Directory for all user and user group information. If you choose to use CA Directory to manage all your user information, it stores all user information (CA SSO-specific as well as general user information).

Administrative Data Store

> This information is stored in CA Access Control. It stores information such as access control lists, resources, authentication hosts.

Types of Users

There are two types of user:

SSO Users

SSO users are people who have the CA SSO Client on their workstations and who access SSO-enabled applications. SSO-specific information about SSO users is always stored in CA Directory which is embedded in CA SSO. General information about SSO users is stored in an LDAP data store such as Microsoft Active Directory or CA Directory.

SSO Administrative Users

SSO Administrative users are people who have administrative privileges to CA SSO and who can make changes to the CA SSO system. Information about SSO Administrative Users is stored in CA Access Control which is embedded in CA SSO.

Note: You must create administrator users in CA Access Control and give those users administrative privileges to CA SSO. You can not use a user that already exists in your user data store.

Active Directory as the User Data Store

If your enterprise already uses Microsoft Active Directory (ADS) as your primary user data store, you should configure CA SSO to use that information.

How ADS is Configured as the Primary User Data Store

Many companies already store their users in Microsoft Active Directory Server (ADS). This section explains how to configure CA SSO to access that user information from an existing ADS user data store.

1. Create a DSA Router to ADS
2. Configure CA Directory to allow the link to ADS
3. Create a user data store on the CA SSO Server to use ADS

Once you have configured ADS as your user data store, we advise you to implement the ADS Listener component, which helps keep information between ADS and CA SSO synchronized.

Create a DSA Router to ADS

To use ADS as your primary user data store, you must create a DSA using the SSO Sever’s embedded CA Directory, to route to ADS. In CA Directory terms, this creates a DXlink between the CA SSO Server and ADS.

This procedure assumes that your ADS computer is called "ADServer01" and that your fully qualified domain name is "acmecorp.com" and your short domain name, or netbios name is just "acmecorp"; the DSA is being created on the CA SSO Server called SSOServer1 in this example.

To create a CA Directory to route to ADS

1. Using Windows Explorer, go to the following directory:

   ```
   C:\Program Files\CA\Directory\dxserver\config\knowledge
   ```

2. Create an empty text file called “AD_ACMECORP_Router.dxc”. Substitute ACMECORP for the AD domain name. This file creates a router DSA called AD_ACMECORP_Router on SSOServer1; it points to the Active Directory AcmeCorp on ADServer1.

   Note: If Windows Explorer is set to hide extensions, the file may incorrectly be created with the extension “.dxc.txt”. This is not correct and you must change the Windows Explorer setup and rename with just the extension “.dxc”.

3. Using notepad, copy the following into the file and change:

   - “ADServer1” to the Microsoft Active Directory computer name
   - “acmecorp” to your domain name
   - "<dc com><dc acmecorp> to your fully qualified domain name (maintaining this format)

   Note: The domain components in the last parameter are in reverse order from usual.

   ```
   # Computer Associates DXserver/config/knowledge
   # AD_ACMECORP_Router.dxc
   # Routes to Active Directory on ACMECORP domain hosted on ADServer1
   # Refer to the CA Directory Administration Guide for the format
   # of the set dsa command.
   set dsa AD_ACMECORP_Router =
   ```

```
{
    prefix            = <dc "com"><dc "acmecorp">
    native-prefix     = <dc "com"><dc "acmecorp">
    dsa-name          = <o AD_ACMECORP><cn AD_ACMECORP_Router>
    dsa-password      = "secret"
    address           = tcp "ADServer1" port 389
    auth-levels       = clear-password, ssl-auth
    dsa-flags         = read-only
    trust-flags       = allow-check-password, no-server-credentials
    link-flags        = dsp-ldap, ms-ad
};
set transparent-routing = true ;
```

Note: The "read-only" dsa-flag prevents updates to AD from the CA SSO Server (even if the account used by the user data store has domain admin privileges).

4. Using notepad, open PS_Servers.dxg and add the following line to the end of the file.

```
source "AD_ACMECORP_Router.dxc";
```

For example:

```
# Computer Associates DXserver/config/knowledge/
#
# PS_Servers.dxg written by CA SSO Server Installation
#
# Description:
# Use this file to group and share DSA knowledge.
# PS DSA's source this file
# from its initialization file.
#
source "../knowledge/PS_ACMECORP.dxc";
source "../knowledge/PSTD_ACMECORP.dxc";
source "../knowledge/AD_ACMECORP_Router.dxc";
```

You must now restart the CA Directory service.

Configure CA Directory to Allow the Link to ADS

To be able to successfully dxlink to the ADS, you must update the CA Directory access controls.

This procedure assumes that your domain name is "acmecorp" and that you have a user object "Prani Patil" stored in the Help Desk Organizational Unit (ou=Help_Desk,dc=acmecorp,dc=com) to be used by the DSA to access the ADS.

To update the CA Directory access controls.

1. Using Windows Explorer, go to the following directory:

   ```
   C:\Program Files\CA\Directory\dxserver\config\access
   ```

2. Open the PS_Access.dxc file in a text editor.
3. Add the following information to the "Define user groups" section at the top of the file:

   ```
   set group = {
   name = "AD_Group"
   users = <dc "com"><dc "acmecorp"><ou "Help_Desk"><cn "Prani Patil">
   };
   ```

   This adds the user ‘Prani Patil’ from the Microsoft Active Directory data store to a group name ‘AD_Group’

4. In the "Grant read and update access for sso-server group in PS and PSTD” section add the following:

   ```
   set admin-user = {
   group = "AD_Group"
   subtree = <dc "com"><dc "acmecorp">
   };
   set admin-user = {
     group = "AD_Group"
     subtree = <o "PS"><ou "LoginInfos"><ou "ad-acmecorp">
   };
   ```

   This configures the CA Directory to allow a connection to read the Active Directory tree (dc=acmecorp,dc=com) as long as the user trying to access it through the CA SSO Server DSA is listed in the AD_Group Access Controls group.This will also allow access to the portion of CA SSO Server’s ps-ldap DSA where application login information objects are stored (ou=ad-acmecorp,ou=LoginInfos,o=PS).

Create a Microsoft Active Directory User Data Store on the CA SSO Server

Even when CA SSO is configured to use ADS as the user data store, it must still keep SSO-specific user information, such as application or logon information, stored on the CA SSO Server in CA Directory.

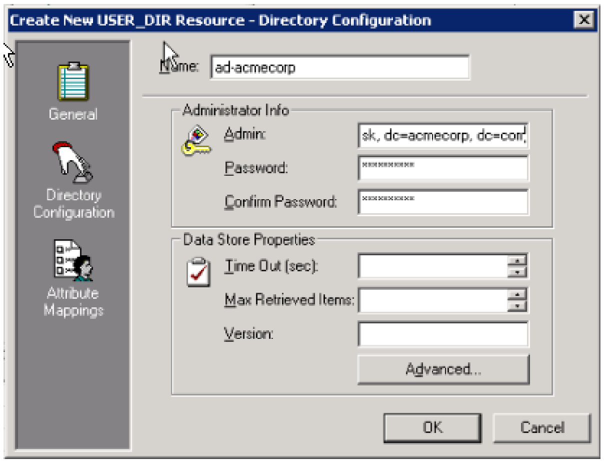
You therefore need to configure CA SSO to get user information from ADS, but get logon and application information from CA Directory on the CA SSO Server.

This procedure assumes that your ADS computer is called "ADServer1", that your domain name is "acmecorp" and that have an employee called Prani Patil who works in the Help Desk department. You must replace this information with information specific for your company.

To create a user data store that points to AD for user records and local LDAP for the user's login variables

1. Log onto the Policy Manager.
2. Go to Resources, Single Sign-On Resources, User Resources, Datastores.
3. Right-click in the right pane and select New.
4. Enter the following in the dialog:
   - Name: ad-acmecorp
   - Data Store Type: AD
   - Owner: [blank]
   - Base Path: dc=acmecorp, dc=com
   - Comment: Active Directory ETRUST Domain Router
   - Host: localhost
   - Port: 13389
5. Click the Directory Configuration icon on left.
6. Configure the datastore using the following dialog. You should use a permanent user, but they do not need to be an administrator. For example,

   Admin: cn=Prani Patil, ou=Help_Desk, dc=acmecorp, dc=com
7. Password: whatever you assigned to this user when creating it.



Create New USER_DIR Resource - Directory Configuration

Name: ad-acmecorp

Administrator Info

Admin: sk, dc=acmecorp, dc=com

Password: **********

Confirm Password: **********

Data Store Properties

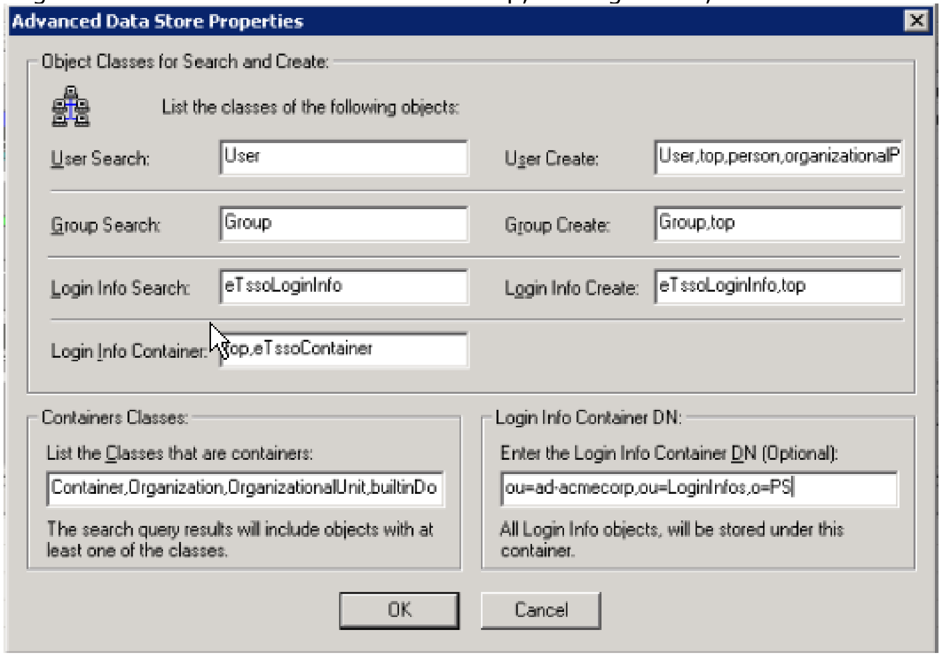
Time Out (sec):

Max Retrieved Items:

Version:

8. Click Advanced on lower right.

Keep all defaults except modify/add the following:
Container Classes:
container,organization,organizationalUnit,builtinDomain,country
Login Info Container DN: ou=ad-acmecorp,ou=LoginInfos,o=PS



Advanced Data Store Properties

Object Classes for Search and Create:

List the classes of the following objects:

User Search: User

User Create: User,top,person,organizationalP

Group Search: Group

Group Create: Group,top

Login Info Search: eTssoLoginInfo

Login Info Create: eTssoLoginInfo,top

Login Info Container: top,eTssoContainer

Containers Classes:

List the Classes that are containers: Container,Organization,OrganizationalUnit,builtinDo

The search query results will include objects with at least one of the classes.

Login Info Container DN:

Enter the Login Info Container DN (Optional): ou=ad-acmecorp,ou=LoginInfos,o=PS

All Login Info objects, will be stored under this container.

Note: You must remove the angle brackets "<" and ">" that may appear in the LoginInfoContainerDN field - these are only here to indicate that you must enter text.

Note: The Containers Classes field determines which classes the Policy Manager interprets as containers. Any typos will cause problems or some containers may not appear in the user data store when viewed with the Policy Manager.

9. Click OK twice to create the user data store.
10. When asked, restart the CA SSO Server service.
11. Go to the Windows Start menu and select Programs, Administrative Tools, Services.
12. Stop the CA SSO Server service.
13. Stop the CA Directory - PS ACMECORP and CA Directory PSTD ACMECORP services.
14. Start CA SSO Server service. This also starts the CA Directory Service.

Configure SSL Communication between the CA SSO Server and the CA Directory and Active Directory Data Store (Windows)

This section describes how to set up an SSL encrypted communication channel between the CA SSO Policy Server and the Domain Controller hosting the Active Directory (AD) to be integrated as a SSO User Data Store.

Communication is accomplished by using the LDAPs (LDAP-Secure) interface provided by the DC (Domain Controller) and utilizing SSL as the communication protocol. Issuing the needed x509-certificates is accomplished by the embedded CA Directory's DXCertGen utility.

Note: It is assumed that the Microsoft Certificate Services are installed and operational on any of the Domain Controllers.

How to Use OpenSSL to Set Up SSL

To use OpenSSL to set up SSL between the CA SSO Server, CA Directory, and AD data store, you must:

1. Download the MS-CA Root Certificate
2. Convert the MS-CA Root Certificate into PEM format
3. Establish the trust between the CA SSO Server and the Domain Controller
4. Create the CA Directory Server DSA Certificates
5. Install the CA Directory ssld service
6. Configure DXlink to utilitize the LDAPs Interface of AD
7. Configure CA SSO Server to utilitize SSL while communicating with AD_userDIR
8. Test and verify.
9. Test an SSL encrypted LDAP connection to Microsoft Active Directory using JXplorer (Optional)

Download the CA Root Certificate

To create the CA Root Certificate

1. On the Domain Controller, download OpenSSL (openSSL.zip) from its community site and unzip the archive to disk.
2. Enable the Certificates MMC Snap In Control from the following location:

   Start Menu > Run > mmc > Console > Add/Remove Snap-in > Add Certificates > Add > Computer Account > Local Computer > Finish > Close > OK
3. Open the MMC and locate the root certificate of the MS-CA in the local computer's Trusted Root Certificates store.
4. Export this certificate to a pfx file (include private key); uncheck "Strong key protection" (make sure to use only the certificate that is letting you export the private key), and save the pfx file in the following locations:

   ```
   ..\certs folder (for example: ..\certs\MS-Root_cert.pfx)
   ```

Convert the MS-CA Root Certificate into PEM Format

To convert the MS-CA Root Certificate into PEM format, open a command prompt, cd to the openSSL folder, and run the following command to convert the pfx file into a pem file:

```
openssl pkcs12 -in ..\certs\MS-Root_cert.pfx -out ..\certs\CAcert.pem -nodes
```

Establish the Trust Between the CA SSO Server and the Domain Controller

To establish trust between the CA SSO Server and the Domain Controller

On the CA SSO Server, map a network drive to the Domain Controller and copy the MS-CA Root Certificate to the embedded CA Directory's Trusted Root Certificates store:

```
copy ..\certs\CAcert.pem "%DXHOME%\config\ssld\CAcert.pem"
```

On the Domain Controller, open a command prompt and import the MS-CA Root Certificate into the CA Directory's Trusted Root Certificates store by issuing the following command:

```
DXCERTGEN -n "%DXHOME%\config\ssld\CACert.pem" importca
```

Create the< CAdir> Server DSA Certificates

Use the following procedure to create the CA Directory Server DSA Certificates.

Open a command prompt and enter the following command:

```
DXCERTGEN certs
```

Install the CA Directory ssld Service

Use this procedure to install the CA Directory ssld service.

From a command prompt on the CA SSO Server machine

1. Run the following command:

   ```
   ssld install caDIRssld -certfiles "%DXHOME%\config\ssld\personalities" -ca
   "%DXHOME%\config\ssld\CAcert.pem"
   ```

   Note: This is one command to be entered in a single line surrounded by "" for path-names containing spaces.

2. Start the ssld service:

   ```
   net start ssld_caDIRssld
   ```

   Note: This task can also be accomplished by starting the "CA Directory SSL daemon - caDIRssld" service from the Services Control panel.

Configure DXlink to Utilize the LDAPs Interface of Active Directory

Use this procedure to configure the DXlink.

1. Open the ad_name_router.dxc file and make sure it contains the following:
   - address = tcp "ADServer1" port 636
   - auth-levels = anonymous, clear-password, ssl-auth
   - link-flags = dsp-ldap, ssl-encryption, ms-ad
2. Edit the PS_<servername>.dxc file:

   auth-levels = anonymous, clear-password, ssl-auth

Configure the CA SSO Server to Utilize SSL While Communicating with the AD_userDIR

From the Policy Manager, edit the ps-ldap and AD user data store and ensure the SSL Connection check box is enabled.

Note: The CA SSO Server and Policy Manager do not exchange any x509-keys so they do not need a keystore themselves, but symmetrically encrypt communication with a random number agreed upon session initiation with the destination.

Test and Verify

To test and very the SSL connection, do the following:

1. Shutdown the CA SSO Server service and the PS and PSTD services and restart accordingly.

   ```
   net stop ssod
   dxserver stop all
   dxserver start all
   net start ssod
   ```

2. Start DXconsole and connect to the Router DSA:

   ```
   telnet localhost 13379
   set trace=x500;
   ```

3. Open the Policy Manager and start browsing the AD_userDIR

   In the PS_<servername>.trace you must find the following sequence:

   ```
   ...

   > <- #4 (SSL) LDAP BIND-REQ

       ...

   > (Remote) -> #5 (SSL) [Router_AD] DXLINK BIND-REQ

       ...

   > (Remote) <- #5 (SSL) [Router_AD] DXLINK BIND-CONFIRM

       ...
   > (Remote) <- #5 (SSL) [Router_AD] DXLINK COMPARE-CONFIRM

       ...
   > -> #4 (SSL) LDAP BIND-CONFIRM

       ...

   > <- #4 (SSL) LDAP SEARCH-REQ

       ...

   > (Remote) -> #5 (SSL) [Router_AD] DXLINK SEARCH-REQ

         ...
   ```

Alternatively, you may also use a network sniffer like Wireshark to verify all communication is handled using SSL.

Test an SSL Encrypted LDAP Connection to Microsoft Active Directory Using Jxplorer (Optional)

(Optional) Use this procedure to test the connection using Jxplorer.

On the SSO Server

1. Make a backup of the current DXserver's offline trusted root CA store:

   ```
   copy "%DXHOME%\config\ssld\trusted.pem" "%DXHOME%\config\ssld\trusted.pem.backup
   ```

2. List the DXserver's trusted root CAs

   ```
   DXCERTGEN listca
   ```

3. Remove all trusted root CAs except the DXCertGenPKI

   ```
   DXCERTGEN -r 1 removeca
   ```

   Note: Repeat until only DXCertGenPKI is listed.

4. Copy the resulting "%DXHOME\config\ssld\trusted.pem" CA Directory-CA Root Certificate and the ..\certs\MS-Root_cert.pfx MS-CA Root Certificate to a temporary location on the JXplorer host.
5. Restore the DXserver's offline trusted root CA store.

On the JXplorer host

1. Enable the Certificates MMC Snap In Control

   Start Menu, Run, mmc, Console, Add/Remove Snap-in, Add, Certificates, Add, Computer Account, Local Computer, Finish, Close, OK

2. Import the copyied trusted.pem file and the MS-Root_cert.pfx into local computer's "Trusted Root Certification Authorities" and "Personal" Certificate Store
3. In JXplorer open "Advanced Keystore Options" from the main menu bar's "Security" option and set the following:

   Set CA/Server Keystore Type: Windows-ROOT

   Set Client Keystore Type: Windows-MY

   The other options are irrelevant, but must not be blank. OK to exit

4. Set up and establish a new connection pointing to the DC on port 636 with Security Level "SSL + User + Password"
5. Set up and establish a new connection pointing to the SSO Server on port 13389 with Security Level "SSL + User + Password" and Base DN o=PS.
6. Set up and establish a new connection pointing to the SSO Server on port 13389 with Security Level "SSL + User + Password" and Base DN pointing to a DN in the AD.

   Note: If there are any issues, make sure time is in-sync on all boxes involved.

Chapter 9: Implementing the ADS Listener

If you use ADS as your primary user data store, you still must store CA SSO-specific user data in CA Directory on the CA SSO Server. This includes authorization rules and logon information. You can keep users synchronized between CA SSO and ADS using the Active Directory Listener. This means that if users are moved or deleted in your Active Directory, the corresponding changes are made to that user's SSO-specific information.

The Active Directory Listener, as the name suggests, listens for changes to user and user group information on ADS and sends notification of these changes to the CA SSO Server, so that the corresponding change can be made to the CA SSO user information.

All login information and authorization rules, stored by the CA SSO Server, are associated with a user or a group using their distinguished name (DN) and their user data store name. When a user or group is moved from one location in the directory (container) to another, their associated rules and login information (for a user) will get lost (since their name has changed) and when a user or group are deleted the above information will still exist. In both cases this information will become “orphaned information” because that information is no longer associated with any user.

The Active Directory Listener service helps to keep the association of this information in CA SSO in sync with the user or group object. It listens to Active Directory for notifications about changes involving users and user groups and notifies the CA SSO Server that those objects were moved or deleted. The CA SSO Server updates or deletes associated application logon information and access rules associated with these user or user group objects.

The Active Directory Listener enables mutual support for typical business scenarios in organizations where an employee is moved from one department (organization unit) to another or where an employee leaves the company.

Note: ADS Listener supports moving users and user groups, but it does not support renaming or moving containers, however, deleting a container is supported.

This section contains the following topics:

Before You Install

This section describes what you need to know before you install the ADS Listener. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Before you install the ADS Listener, you should install your CA SSO Servers. You must know the names of the computers these have been installed on before you begin this process.   You should also have set up your CA SSO Servers to point to your ADS as a user data store.

Decide Where to Install the ADS Listener

The ADS Listener:

- Can optionally be installed on the Domain Controller
- Must be installed on a domain member
- Must not be installed on the CA SSO Server host

Although the ADS Listener can be installed on any machine, the following list describes the pros and cons of some of the options.

Active Directory Domain Controller (DC) machine

In a single domain controller environment, we recommend that you install the ADS Listener on the DC because this saves network traffic.

A domain member machine

In a multiple domain controller environment, we recommend installing the ADS Listener on a domain member that is not the domain controller because this improves failover.

CA SSO Server host

We do not recommend installing the ADS Listener on the same machine as the CA SSO Server, because this reduces CA SSO Server performance: the ADS Listener and the CA SSO Server share the same network interface, overloading it with the additional network traffic coming from the Active Directory DC.

Wizard Installation versus Silent Installation

There are two ways to install the ADS Listener:

- Wizard installation (Windows GUI)
- Silent installation (command line prompt)

If you choose to do a silent install you must specify the variables by either:

- Creating a response file
- Using command line parameters

Note: The Windows command line may limit the number of characters in a single command. For complex or detailed command line installations, you should consider using a response file.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the ADS Listener. Ensure you have:

- Met all system requirements before you install the ADS Listener. For information about supported platforms and system requirements, see the *SSO Readme* file.
- The ADS host name
- The administrator name and password for the ADS Listener host
- The ADS user data store name
- The ADS user data store base path
- The ADS user data store monitoring context (multiple monitoring contexts supported)
- The ADS naming context
- The CA SSO Server names
- The administrator name and password for the CA SSO Server host computer (you created this when you installed the CA SSO Server)

Install the ADS Listener

This section explains how to install the ADS Listener.

Install the ADS Listener using the Wizard

Note: When you enter more than one computer name in a list, such as all the CA SSO Servers in a server farm, you can separate the names using either a comma or a space.

To install the ADS Listener using the wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically appears. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Active Directory Listener

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install the ADS Listener using Silently

You can install the ADS Listener silently. This means that you provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the ADS Listener at the bottom of the license agreement.

Note: The Windows command line may limit the number of characters in a single command. For complex or detailed command line installations, you should consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically appears. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select ADS Listener.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen. This is required for all silent installations.

   You can now install the ADS Listener using silent installation.

4. Open a command prompt and navigate to the ADS Listener folder on the product DVD.
5. From the command prompt, type:

   ```
   msi.exe   /i "CA SSO Active Directory Listener.msi" /qn {options}
   ```

   options

   Specifies the options to include in the silent install. For more information on command line options, see the following section.

msiexec - Silent Installation Program

The command line parameters for silently installing the ADS Listener include the following options:

LDAP_HOST

Specifies the name of the ADS host

LDAP_PORT

Specifies the ADS LDAP port

Value: Port number

Default: 389

SEARCH_NAMING_CONTEXT

Specifies the Active Directory naming context.

LDAP_ADMIN

Specifies the Active Directory administrator name.

LDAP_PASSWORD

Specifies the Active Directory administrator password.

PS_SERVER

Specifies the CA SSO Server name(s).

AD_NAME

Specifies the Active Directory user data store name.

PS_ADMIN

Specifies the CA SSO Server LDAP administrator name.

PS_PASSWORD

Specifies the CA SSO Server LDAP administrator password.

DATA_STORE_BASE_PATH

Specifies the user data store base path.

MCREGISTRYLIST

Specifies the list of Monitoring Contexts specified in the format of [monitoring context 1] [monitoring context 2] and so forth.

COMM_MODE

Specifies the Communication mode between ADS Listener and CA SSO Server:

Values: 0 | 1 | 2

- 0 – Compatible Mode
- 1 – FIPS-Only TLS Mode of Communication
- 2- TLS Mode of Communication

IDENTITYFILECTRL

Specifies the path to the Certificate file (.pem file)

PRIVATEKETCTRL

Specified the path to the Certificate Key file (.key file)

Example Silent Installation

Here is an example of a silent installation command.

```
msiexec /i "CA SSO Active Directory Listener.msi"  /qn /l*v log.log LDAP_HOST=dcmachine LDAP_PORT=389
SEARCH_NAMING_CONTEXT= dc=domain,dc=com
LDAP_ADMIN=cn=administrator,cn=users,dc=domain,dc=com LDAP_PASSWORD=password
PS_SERVER=psmachine AD_NAME=ad-ldap PS_ADMIN=ps-admin PS_PASSWORD=password2
DATA_STORE_BASE_PATH=ou=org,dc=domain,dc=com
MCREGISTRYLIST=[ou=eTrust,dc=xena-dev,dc=ca,dc=com][ou=Australia,ou=eTrust,dc=xenadev,dc=ca,dc=co
m]
```

Configure the ADS Listener

This section explains the configuration settings for the ADS Listener.

Active Directory Server data is configured as follows:

[HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\SingleSignOn\ADS Listener\AD]

Port

Defines the Active Directory LDAP port number

Value: *port number*

Default: 389

Host

Defines the Active Directory Domain Controller host name or a list of names (comma or space separated)

Value: *computer name*

ADNamingContext

Defines the Active Directory domain DN (for example: dc=domain, dc=com).

Value: *domain name*

UseSSL

Defines whether to use LDAP over SSL while communicating with Active Directory.

1 = SSL enabled

0 = SSL diabled

Value: 0|1

FailureWaitInterval

Defines the time (in seconds) to wait between each connection attempt in case of a problem connecting to Active Directory.

Value: *time in seconds*

Default: 20000

MonitoringContext_X

Specifies the DN of the container to be monitored. This DN should be the same as the Base Path (defined in the SSO Server's Active Directory User Data store) or a location below that. ADS Listener can monitor multiple contexts. X is a sequential number (starting from 0) of the defined monitored context.
For every one of those keys a MonitoringContextScopeSubTree_X can be specified to determine if that Monitoring Context should be treated as a sub tree scope (scope that includes also its sub containers) or one level only.

For example:
MonitoringContext_0=ou=eTrust,dc=xena-dev,dc=ca,dc=com
MonitoringContextScopeSubTree_0=0
MonitoringContext_1=ou=Australia,
ou=eTrust,dc=xena-dev,dc=ca,dc=com
MonitoringContextScopeSubTree_1=1

(The above defines the parent container as a one level container and one of its child-containers as a sub-tree scope. Note that those contexts don't overlap.)

Value:

1=sub tree

0=one level

Default: 1

SSO Server data is configured as follows:

[HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\SingleSignOn\ADS Listener\PolicyServer]

Host

Defines the SSO Server host name (or a list of names). Host names can also be in a host:port format.

Value: *computer name* or *computer name:port number*

ADUserDataStore

Defines the Active Directory user data store name as defined in the SSO Server.

Value: *data store name*

BasePath

Defines the Active Directory base path as defined in the user data store in the SSO Server.

Value: *path*

KeyFilePath

Defines the full name, including path of the file, that will be used to store SSO Server's public key information. This file is created automatically.

Value: *path and file name*

MsgPath

Defines the directory path to where SSO Server's message file is located.

Value: *path*

MsgFile

Defines the SSO Server's message file name ("enu.msg").

Value: *file name*

ADS Listener generic information is configured as follows:

[HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\SingleSignOn\ADS Listener\Main]

Logger

Defines the full name, including path, to the ADListenerLog.ini file.

Value: *path and file name*

DataPath

Defines the path to the data directory, containing the two encrypted credentials files (ads_ps.dat containing SSO Server credentials and ads_ldap.dat containing Active Directory credentials).

Value: *path*

MaxADQueueSize

Defines the maximum number of Active Directory notifications that will be stored in Active Directory queue while waiting to be processed by the notification handler. In the event that this queue reaches its limit, a message will be written to the log file, saying: "Fail to submit XXX Notification to queue. AD Queue is full."

Value: *number of notifications*

Default: 250

MaxPSQueueSize

Defines the maximum number of SSO Server notifications that will be stored in SSO Server queue while waiting to be sent to SSO Server. In the event that this queue reaches its limit, a message will be written to the log file, saying: "Fail to submit XXX Notification to queue. PS Queue is full."

Value: *number of notifications*

Default: 250

OfflineDataPath

Defines the path to the OfflineData directory, where all offline information is stored.

Value: *path*

CommMode

Defines the communication mode.

Values: [0 | 1 | 2]

- 0 – Compatible Mode
- 1- FIPS Only TLS Mode
- 2- TLS Mode of Communication

Cert_filepath

Defines the path to the Certificate file (.pem) file.

Key_filepath

Defines the path to the Certificate key file (.key) file.

The Installation will create two encrypted files that contain the credentials required in order to connect Active Directory and the credential required to connect to the SSO Server. This information can be changed by running:

ADListener -s <user_name> <password>

This is used to set the user name and password for SSO Server.

ADListener -l <ldap_user_dn> <password>

This is used to set the user name and password for Active Directory.

The ADS Listener service may be installed on any computer in the network. It can remotely monitor Domain Controllers. It can only listen to one Domain Controller at a time. When multiple Domain Controllers are involved, the ADS Listener will listen to one Domain Controller at a certain time, with the ability to failover to one of the others.

Chapter 10: Implementing Authentication

This section contains the following topics:

About SSO Authentication

The process by which end users identify themselves to CA SSO is called primary authentication. You can implement different types of security software and hardware which let users perform primary authentication in different ways.

CA SSO comes with native SSO authentication. In addition, CA SSO provides authentication agents to let you use a number of third-party authentication methods. This chapter describes how to implement each of the authentication agents that are supplied with CA SSO:

- Certificate
- LDAP
- RSA SecurID
- Windows

To provide operational flexibility, authentication agents serve as a bridge for communication between the CA SSO Client and the authentication server.

How Primary Authentication Works

The following steps describe the primary authentication process:

1. The user starts the CA SSO Client on their workstation.
2. The CA SSO Client checks the auth.ini file for the list of server sets and authentication methods.
3. The authentication dialog appears, prompting the user to:
   - Select the appropriate server set
   - Provide their credentials, such as a user name and password, biometric information, or a certificate and its passphrase, which may be in the local certificate store or on a smart card.
4. The CA SSO Client passes the user's credentials to the authentication software specified in the auth.ini configuration file, for example, RSA, LDAP, Certificate or Windows.
5. The authentication software verifies that the credentials are valid, either using its own built-in mechanism or (the more common scenario) by sending them to an authentication host with a verification request.
6. If the credentials are successful, the authentication agent creates an SSO ticket, encrypts it using a configured encryption key, and sends it to the CA SSO Client. The SSO ticket includes user identification, authentication method, and time stamp. The ticket is valid for a defined number of hours.

   If the credentials are not valid, the authentication agent sends an error message to the CA SSO Client, informing it that the primary authentication request has failed.
7. The CA SSO Client sends the SSO ticket as a login request to the SSO Server.
8. The SSO Server verifies the SSO ticket.
9. If the ticket is valid, the SSO Server retrieves from the user data store the list of the applications that the user is authorized to use, and sends the list to the CA SSO Client.

   If the ticket is not valid, logon fails and the user receives an error message.
10. The CA SSO Client displays the list of applications. The user can now start work.

Offline Authentication

Offline authentication refers to whether the CA SSO Client can log onto CA SSO and access SSO-enabled applications when there is no connection between the client and the network, or if the authentication agent or authentication host is unavailable.

When offline access is enabled the CA SSO Client caches the following information:

- Authentication credentials
- Application list
- Application script
- Logon variables
- Timestamp of the cache

Authentication Credentials Caching

If offline operation is enabled, the valid user logon details are encrypted and stored on the user's computer. This means that when a user logs onto CA SSO when the client is not connected to the network, or when the authentication host or server is down - they can still be authenticated and run their applications. The next time the client connects to the network (or the auth agent/host is available again) the cached details are checked and validated.

Authentication List Caching

Every time a user logs onto the CA SSO Client on the network it checks with the SSO Server to see if the user's login variables for any of their offline applications have changed. If it has, the CA SSO Client downloads the latest information. This means the user is able to run selected applications when they use CA SSO offline.

Application Script Caching

If an application is configured for offline use, the CA SSO Client caches its script in the Cache Directory. As application scripts may be shared between users, this directory is in a global location available to all CA SSO Client users.

On all subsequent connections, the CA SSO Client checks to ensure all script information is up-to-date. If a script for an application marked for offline use is not present, the CA SSO Client downloads the missing script. If a script is out-of-date, the CA SSO Client queries the timestamp information and determines if a new download is required.

If all information is up-to-date, the CA SSO Client performs no further requests of the server. This helps to save on network bandwidth between the server and client.

Log In Variable Caching

In addition to application script caching, if an application is marked for offline use, the CA SSO Client retrieves the logon variables for that user for that application and caches them in encrypted form in the user-specific Cache Directory.

Before You Begin

This section guides you through what you need to know before you implement authentication. In addition to the information in this section, make sure you review your implementation plan and take note of any specific requirements.

Decide Which Authentication Method to Use

Decide which authentication method to use in conjunction with your IT security manager.

- If you already have an authentication method deployed in your organization you may wish to continue using it.
- If you wish to use biometric authentication you may already have third-party software and wish to continue to use that software.
- If you do not want to use your existing authentication methods, or have none deployed, you can use the CA SSO authentication method, or create a custom authentication agent to use with CA SSO.

Decide Where to Install the Authentication Agent

Here is a list of all authentication methods supported by CA SSO and their preferred installation locations.

LDAP authentication agents

If you want to implement LDAP authentication, you must install the LDAP authentication agent on a computer with a TCP/IP connection to the CA SSO Client computer, and with an appropriate connection to the relevant authentication server, such as Microsoft Active Directory.

You can set up LDAP authentication with most LDAP based directories. LDAP authentication can be configured during installation to work with Microsoft Active Directory or CA Directory.

SSO authentication

You do not need to install an authentication agent for SSO authentication because the SSO Server verifies the user credentials and creates a ticket.

Note: SSO Authentication does not support hierarchical name spaces.

Windows authentication agents

You should install the Windows authentication agent on a domain controller, or a machine that is on the domain.

Note: If this is not possible, you need to install it on a machine which is member of that domain.

Certificate authentication agents

If you want to implement Certificate authentication, you must install the authentication agent on a computer with a TCP/IP connection to the CA SSO Client computer, and with an appropriate connection to the computer where the relevant authentication server software is deployed.

RSA SecurID authentication agents

If you want to implement RSA SecurID authentication, you must install the authentication agent on a computer with a TCP/IP connection to the CA SSO Client computer, and with an appropriate connection to the computer where the relevant authentication server software is deployed.

Note: For more information on system requirements, see the *SSO readme* file.

Design Your Server Sets on the CA SSO Client

To use any of the authentication methods for primary authentication, you must define server sets by correctly configuring the auth.ini configuration file.

More information:

Implementing the CA SSO Client (see page 125)

Synchronize Operating Systems

All operating system clocks must produce a reliable and correct timestamp for the time-zone where each computer hosting CA SSO components is located. For example, a computer located in New York hosting a CA SSO Server will have its operating system clock set to US Eastern Daylight Time (EDT), and a computer located in San Francisco hosting a LDAP Auth Agent will have its operating system clock set to US Pacific Daylight Time (PDT).

Pre-Installation Checklist

Use this checklist to ensure you have performed all pre-installation tasks before you install the authentication agent:

- Ensure that all system requirements are met before you install the authentication agent.

  For more information, see the *SSO readme* file.
- Ensure that your CA SSO Server(s) has been installed and configured.
- Ensure that you know the host name of the computer or computers on which you are installing the authentication agent.
- Ensure that the computer on which you are installing the agent has TCP/IP communications with the CA SSO Client computers.
- Ensure all operating system clocks produce a reliable and correct timestamp for the time-zone where each computer hosting any SSO components is located.

More information:

Synchronize Operating Systems (see page 246)

Implement Certificate Authentication

CA SSO supports primary authentication using PKCS#12 certificates. This section provides an overview on key certificate authentication concepts, and explains how to install and configure the certificate authentication agent.

Revocation Settings

Certificate Revocation List (CRL) is a list identifying revoked certificates and is signed by a Certificate Authority (CA). They are used to manage compromised private keys, or when the right to authenticate with a certificate is lost during the certificate's validity period. There may be several reasons for this, but in both cases, the certificate needs to be revoked.

There are several ways that the system can identify revoked certificates:

CRL

This is a list of certificates that have been revoked by the Certification Authority. The CRL is a blacklist that contains the certificates which are no longer valid.

The CRL file can be:

- On the local machine on which the Certificate authentication agent is installed.
- On a remote machine share.
- Hosted on a HTTP web server such as Microsoft IIS
- Hosted on an LDAP server

Fixed OCSP

Fixed Online Certificate Status Protocol (OCSP) lets you specify a fixed address for an OCSP responder that can check the user certificates and verify whether they are valid or have been revoked.

You also must have the full address (Hostname/IP address and the port) of the responder to use this option.

AIA OCSP

AIA OCSP lets the Certificate authentication agent retrieve the OSCP responder address from the user certificate. This means that you do not have to specify a fixed OCSP address. To use this option the users' certificates must contain an OCSP responder address in the 'Authority Information Access (AIA)' attribute of the certificate.

CRL DP

The CRL DP stands for CRL Distribution Points. This option lets the Certificate authentication agent retrieve a CRL using either HTTP or LDAP by using the first address listed in the 'CRL Distribution Points' attribute of the user's certificate.

You also must have the issuing/signer certificate of the CRL(s) to be used by the Certificate authentication agent.

You must specify at least one issuing/signer certificate. These certificates must reside in the same directory.

Revocation Combinations

The Certificate authentication agent lets you use a combination of two of the available Revocation Status Checking Methods. All combinations consist of CRL together with another method. The available combinations are:

- CRL and Fixed OCSP
- CRL and AIA OCSP
- CRL and CRLDP

The benefit of using a combination of Revocation Status Checking Methods is that it provides a more accurate result, thus providing more security. The Certificate authentication agent always first checks the certificate with the CRL. If the certificate is listed as revoked here, the authentication agent does not check the second method. If the certificate is not listed as revoked on the CRL, the authentication agent checks the second method, and the result of the second method is returned. The configuration for each of the methods is the same as if you selected them individually.

CRL Expiry Mode and Grace Period

CRL Expiry Mode specifies what action to take when the existing CRL has expired. It applies to both CRL and CRL DP revocation settings and can be set to the following values:

Ignore CRL next update

Ignore the next update time of the CRL and use the existing CRL to check certificates.

Ignore CRL next update but log error message when grace period expires

Includes *Ignore CRL next update*. After Next Update + Grace Period, log error message when using expired CRL to check certificate status.

Fail to authenticate after CRL next update + grace period

Within Next Update + Grace Period, log error message when using the expired CRL to check certificate. After Next Update + Grace Period, fail the authentication attempt. Default to mode 3.

Note: The CRL expiry mode and grace period are set during installation. These settings can be modified post-installation in the Certificate authentication agent's configuration file CA_certtga.ini file, but you must restart the TGA service for the changes to take effect every time you modify the INI file.

Name Mapping

When a Certificate authentication agent has verified that a user certificate is valid, it creates an SSO ticket for that user. The SSO ticket is sent back to the CA SSO Client which uses the ticket to log the user on to the CA SSO Server.

To identify which user the SSO ticket belongs to, the SSO ticket contains a user name field that identifies the user name. The value must match a corresponding value or attribute for that user in the CA SSO Server user data store.

By default, this value is set to the Common Name (CN) attribute during the Certificate authentication agent installation process. This value can be changed to another attribute post-installation using the CA_certtga.ini file.

Note: You must restart the TGA service for the changes to take effect every time you modify the Certtga.ini file.

More information:

Configure a New Name Mapping Method (see page 262)

Name Mapping Settings

You can change the default name mapping method using the *MappingMethod* setting in the CA_certtga.ini file. You can use any of the Certificate attributes listed in the second table to populate the *MappingMethod* field. It is worth noting, however, that values such as C (country) or O (organization) are not user-specific and are therefore generally unsuitable for populating the user name field in the SSO ticket.

Name	Type	Data
MappingMethod	REG_SZ	CN
NameMappingDLLPath	REG_SZ	C:\Program Files\CA\CA Single Sign-on\Certificate Agent\name_mapping.dll **Note:** If you are upgrading from a previous version of SSO, the path is: C:\Program Files\CA\eTrust SSO\Certificate Agent\name_mapping.dll

Attribute Code	Attribute	Location in Certificate
CN	Common Name	Subject DN
DN	Distinguished Name	Subject DN
OU	Organizational Unit	Subject DN
C	Country	Subject DN
O	Organization	Subject DN
L	Location	Subject DN
EMAIL	Email address	Subject Alternative Name
IP	IP Address	Subject Alternative Name
DNS	DNS	Subject Alternative Name
URI	URI	Subject Alternative Name

Note: The default DLL only extracts the first instance of an attribute. For example, if a certificate contains two OU fields, only the first encountered are extracted.

Certificate/Key Storages

The Certificate authentication agent supports three types of certificate/key storages for the CA SSO Client user:

PKCS#11 Tokens

PKCS#11 is accessed through the Application Programming Interface defined in the PKCS#11 standard. Only PKCS#11 tokens that are supported by the chosen PKCS#11 library can be used. The keys on a PKCS#11 token are usually protected by a PIN number. Some PKCS#11 tokens also support Protected Authentication Path (PAP). PAP is implemented using a fingerprint reader where the user's fingerprint is used to access the keys stored.

The client can be configured to force the user to use PAP when the PKCS#11 token involved supports it.

PKCS#12 Files

The user's certificate and private key information is stored in a PKCS#12 file. The PKCS#12 file can be password protected.

MSCAPI Certificate Stores and Smart Cards

Windows platform users can use the *"My" store* to store certificate and keys. Before the certificate and keys can be used, they must be imported into the *"My" store*.

Certificate and keys stored on MSCAPI enabled smart cards can also be used.

Note: The certificate that can be used for authentication must include the "Key Usage extension" with the "*Digital Signature*" bit set.

Before You Install

This section describes what you need to know or do before you install the Certificate authentication agent. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Configure the CA SSO Client for Certificate Authentication

Use the Auth.ini file to configure the user's Certificate authentication settings, including:

- Authentication agent server name and port information
- Authentication method

To configure the CA SSO Client for Certificate authentication

1. Open the Auth.ini file.
2. Add CERT to the *AuthMethods* setting, for example:

   ```
   [ServerSet1]
   AuthMethods=CERT LDAP SSO
   ```

3. Add the Certificate authentication agent server name to the *AuthCERT* setting. For example:

   ```
   [ServerSet1]

   AuthCERT=ssoaa-a
   ```

4. (Optional) Configure the Certificate authentication agent server port number. For example:

   ```
   [ServerSet1]

   AuthCERT=ssoaa-a:13987
   ```

   If the port number is not specified, the default port (13987) is used.

5. Specify the values of the other settings associated with Certificate authentication in the [Auth.CERT] section of the Auth.ini file. For example:

   ```
   [Auth.CERT]
   CertStore=FILE
   Pkcs11LibraryPath=<FULL_PATH_TO_PKCS11_LIBRARY_FILE>
   Pkcs11PromptText=
   DisablePasswordField=no
   AutoAuthenticate=no
   CertThumbprint=
   DefaultServerSet=ServerSet1
   ```

   Note: The PKCS11 library path has to be specified if PKCS11 (Smartcard) needs to be selected as source of certificates during or after installation.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the authentication agent:

- Ensure that all system requirements are met before you install the authentication agent.

  For more information, see the *SSO readme* file.

- Ensure your CA SSO Servers have been installed and configured.
- Ensure that the computer on which you are installing the agent has TCP/IP communications with the CA SSO Client computers.

- Ensure you know all relevant information prior to running the installation including:
  - The CRL file and its location.
  - The certificate of the CA that issued the CRL, and its location.
  - The CRL checking method combinations. For example, None, CRL, or a combination of CRL and one of Fixed OCSP, AIA OCSP, or CRLDP.
    - For Fixed OCSP and AIA OCSP checking methods, you can provide the certificate that is used to sign the OCSP request to your responder. This must be a PKCS#12 file. You must also provide a password. You can also choose not to sign the OCSP requests that are sent from the authentication agent to the OCSP responder service, in which case, you do not need to specify a signing certificate at all. However, for security reasons, we recommend that you specify one instead of choosing not to sign requests.
    - For CRL and CRLDP checking methods, you must provide the time interval between each poll for an updated CRL.
  - CRL expiry mode and grace period.
  - HTTP proxy configuration if it is being used to retrieve the CRL over HTTP or to interact with responder service for Fixed or AIA OCSP methods.
- Ensure that you know the type of name mapping you want to implement. By default, this is set to the Common Name attribute during the installation process. You can change this using the CA_certtga.ini file post-installation.
- For silent installs, you must first run the Certificate authentication agent wizard and read the license agreement. The command line option for accepting the license agreement can be found at the bottom of the license agreement text.
- Ensure all operating system clocks produce a reliable and correct timestamp for the time-zone where each computer hosting any SSO components is located.
- If you want to configure TLS/SSL communication (FIPS communication) between the CA SSO Client and the Certificate authentication agent, you need to make sure that all resources used, such as certificates, CRL(s), and so forth, are FIPS 140-2 compliant. If the user certificates do not use FIPS 140-2 compliant algorithms, they will not work for TLS/SSL communication. The installation DVD contains a folder with sample user certificates that are FIPS 140-2 compliant.

Trusted Certificates

The Certificate authentication agent uses a list of trusted certificates which are traceable up to the Certificate Authority (CA) to determine the validity of a user certificate. Unless the path to the issuing CA of the user certificate is included, the Certificate authentication agent cannot verify the user certificate.

When you install the Certificate authentication agent you will be asked to specify a trusted certificate. You can use a 'Browse' button to navigate to the directory that contains the DER-encoded certificate.

You must specify at least one trusted certificate to install the Certificate authentication agent, but you may also specify multiple trusted certificates. These certificates must all be located in the same directory.

Install the Certificate Authentication Agent

To install the Certificate authentication agent you must create and install the necessary trust certificate files, and then install and start the Certificate authentication agent service.

This section explains how to install the Certificate authentication agent.

Install using the Wizard

Before you install the Certificate authentication agent you should have all your certificates saved in a single directory on the computer on which you intend to install the agent. This directory should contain at least one trusted certificate. You will be prompted for the certificates during installation.

To install the Certificate authentication agent

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer wizard expand the CA SSO Authentication Agents folder, and select Certificate Authentication Agent.

   The Install button becomes active.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install Silently

You can install the Certificate authentication agent silently. This means that you need to provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the Certificate authentication agent at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Certificate Authentication Agent.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the Certificate authentication agent using silent installation.

4. Open a command prompt and navigate to the Certificate authentication agent folder on the product DVD.

5. From the command prompt, type:

   ```
   setup.exe -silent -V LICENSE_VIEWED=value {parameters}
   ```

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

Setup Command—Install Certificate Authentication Agent

The command line parameters for installing the Certificate authentication agent include the following options:

-P installLocation

Defines the install location.

The command has the following format:

-P installLocation=[*Value*]

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

-silent

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

-V IS_REBOOT_NOW=[*Value*]

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V CrldpIssuerCertPath

Specifies the location of .DER encoded CRL certificate files.

The command has the following format:

-V CRLDPISSUERCERTPATH=[*Value*]

Value: The location of the CRL file(s). Each path surrounded by quotation marks if the path contains spaces.

-V CrldpIssuerCerts

Specifies the list of the DER-encoded CRL certificate files.

The command has the following format:

-V CRLDPISSUERCERTS=[*Value*]

Value: The list of .DER encoded issuer certificate files.

-V CrlRetrievalTimeout

Specifies the timeout for retrieval of CRL or CRLDP revocation. The minimum is for 30 seconds.

The command has the following format:

-V CrlRetrievalTimeout=[*Value*]

Value: The timeout period in seconds for retrieval of CRL/CRLDP revocation information.

-V CrlFileName

Specifies the Certificate Revocation (CRL) file. It must be signed by a CA and be DER-encoded. This may be a local file, remote file, HTTP, URL, or LDAP URL.

The command has the following format:

-V CRLFILENAME=[*Value*]

Value: Defines the name of the .DER encoded file.

-V CrlIssuerCert

Specifies the CA certificate that issued the CRL.

The command has the following format:

-V CRLISSUERCERT=[*Value*]

Value: The name of the CA certificate in .DER format that issued the CRL.

-V CrlPollInterval

Specifies the CRL Polling Interval in seconds. That is, how often to poll for updates of the CRL. If this is 0, there will be no polling for new CRL updates.

The command has the following format:

-V CRLPOLLINTERVAL=[*Value*]

Value: The time in seconds to check for an updated CRL.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

-V LICENSE_VIEWED=[*Value*]

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V OcspResponder

Specifies the http://hostname-of-ocsp-responder:port-number-of-responder. Default port is 3080.

The command has the following format:

```
-V OcspResponder=[Value]
```

Value: The HTTP address and port number for the OCSP responder. For example, http://ssoca-a:3080 (or) https://ssoca-a: 3443.

-V OcspSignCert

Defines the Certificate with which to sign OCSP Requests. Must be a PKCS#12 file.

The command has the following format:

```
-V OCSPSIGNCERT=[Value]
```

Value: The name of the signed certificate used to sign OCSP requests.

-V OcspSignCertPass

Defines the password/passphrase for the OSCP signed certificate.

The command has the following format:

```
-V OCSPSIGNCERTPASS=[Value]
```

Value: The password/passphrase for the OCSP signed certificate.

-V HttpProxy

Specifies the proxy address to access an OCSP Responder or retrieve CRL over HTTP. "IE5://" specifies to use the Internet Explorer settings on the system.

The command has the following format:

```
-V HttpProxy=[Value]
```

Value: The proxy address. For example, http://ssoproxyca-a:8080.

-V TrustedNames

Defines the list of .der/.crl files.

The command has the following format:

```
-V TrustedNames=[Value]
```

Value: The list of .der/.crl files.

-V TrustedCertDirectory

Defines the location of .der/.crl files.

The command has the following format:

-V TrustedCertDirectory=[*Value*]

Value: The location of the files.

-V MaxCertChainDepth

Specifies the maximum depth of the certification chain. The default value is 2.

The command has the following format:

-V MaxCertChainDepth=[*Value*]

Value: Depth of the specification chain.

-V RevocationMeth

The command has the following format:

-V RevocationMeth=[*Value*]

-V CrlExpiryMode

The command has the following format:

-V CrlExpiryMode=[*Value*]

-V CrlGracePeriod

The command has the following format:

-V CrlGracePeriod=[*Value*]

-V AuthHostNameValue

The command has the following format:

-V AuthHostNameValue=[*Value*]

-V TicketEncryptionKeyValue

The command has the following format:

-V TicketEncryptionKeyValue=[*Value*]

-V AGENT_ID_FILE

Defines the Trusted Certificate file in .pem format.

The command has the following format:

-V Agent_ID_FILE=[FilePath]

FilePath: Indicates the absolute path to the .pem file (along with file name)

-V AGENT_KEY_FILE

Defines the Trusted Certificate file in .pem format.

The command has the following format:

```
-V AGENT_KEY_FILE =[FilePath]
```

FilePath: Indicates the absolute path to the .key file (along with file name)

-V COMM_MODE

Defines the mode of communication.

The command has the following format:

```
-V COMM_MODE=[Value]
```

Value: 0 | 1 | 2.

- 0 - Non-FIPS mode
- 1 - FIPS-only
- 2 - Mixed mode

Install Using Silent Installation and Response File

Use the following procedures to install the Certificate authentication agent silently using a response file.

Note: You can use a combination of response file and command line options to silently install the Certificate authentication agent. In this case, the command line options override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the Certificate authentication agent folder on the product DVD.

3. From the command prompt, type:

   ```
   setup.exe -silent [parameters] -options {response file}
   ```

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create a Certificate Authentication Agent Response File (see page 261)
Setup Command—Install Certificate Authentication Agent (see page 256)

Create a Certificate Authentication Agent Response File

Response files contain user specified install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the Certificate authentication agent folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record {file name}
   ```

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.
3. Complete the installation.

   A response file is created in the directory specified.

Post-Installation Configuration Options

The following sections explain some of the post-installation configuration options.

Configure CA_certtga.ini Settings

Once you have installed the authentication agent, you can configure the CA_certtga.ini settings file with any post-installation changes.

Create an Authentication Host Entry on the CA SSO Server

Note: This procedure is only required if you decide to change the default authentication host information.

To create an authentication host entry on the CA SSO Server

1. Define a new authentication host on the CA SSO Server.
2. Update the CA_certtga.ini file to include the new information for the *TicketKey* and *AuthHostName* settings. For example:

   ```
   [Ticket]

   TicketKey=32456164

   AuthHostName=CERT_Authhost
   ```

   Ticket encryption keys have a maximum length of 256 characters.

Configure a New Name Mapping Method

Use this procedure to change the name mapping method used during user authentication with the CA SSO Server.

By default, the name mapping method value is set to the Common Name (CN) attribute during the authentication agent installation process.

To configure a new name mapping method

1. Open the Certificate authentication agent configuration file CA_certtga.ini.
2. Update the file to include the new information for the *MappingMethod* setting. For example:

   ```
   [NameMapping]

   MappingMethod=EMAIL

   NameMappingDLLPath=<FULL_PATH_TO_NAME_MAPPING_DLL
   ```

Configure Custom Name Mapping

You can customize your own user name identifier by creating a custom name mapping DLL.

- To use a user attribute in the certificate file that is not listed in the default name mapping attributes table, you must create a custom DLL with the following exported functions:
  - Etsso_Name_Mapping_RC etsso_certtga_NameMappingGetName(const unsigned char* cert, const unsigned int len,unsigned int* buffLen, wchar_t* nameBuff);
  - Etsso_Name_Mapping_RC etsso_certtga_NameMappingInit(void);
  - void etsso_certtga_NameMappingTerm(void);
- To use the custom name mapping DLL, specify the path to the custom DLL in the "NameMappingDLLPath" setting in the CA_certtga.ini file. The Certificate authentication agent uses this entry to load the DLL.

  The Certificate Auth Agent uses the nameBuff it gets back from the call to etsso_certtga_NameMappingGetName as the username when it creates the SSO Ticket.

Configure Certificate Authentication to Work with Active Directory

This section explains how to configure the Certificate authentication agent to retrieve a CRL from an Active Directory using a CRLDP from the client certificate.

It covers:

1. Installing the MS Windows support tools on the Active Directory computer
2. Creating an ADSI edit console
3. Enabling anonymous access to Active Directory
4. Enabling anonymous access to the CRL store in Active Directory
5. Defining the CRLDP address that will be published in the user certificates
6. Configuring the Certificate authentication agent to use the CRLDP address from the user certificate

Step 1: Install the MS Windows Support Tools on the Active Directory Computer

From the Windows 2003 Server CD install the SUPTOOLS.MSI found in the directory \SUPPORT\TOOLS\

Step 2: Create an ADSI Edit Console

Use the following steps to create an ADSI edit console.

To create an ADSI edit console

1. Select Start, Run, and enter mmc.

   The Microsoft Management Console application starts.

2. Select File, Add/Remove Snap-in
3. Click Add.

   A list of available snap-ins displays.

4. Select ADSI Edit from the list and click Add.

   ADSI Edit is added to the list of snap-ins.

5. Click Close and then OK.
6. Right-click on the "ADSI Edit attribute" under the Console Root entry and select Connect to.
7. Make sure the Select a well known Naming Context option is enabled.
8. Select Domain from the drop down list and click OK.
9. Repeat steps 6-8, this time selecting Configuration from the context menu.
10. Click OK.

Step 3: Enable Anonymous Access to the Active Directory

Use the following steps to enable anonymous access to Active Directory.

To enable anonymous access to Active Directory

1. From within the ADSI Edit console, expand the Configuration node and navigate to the following location:
   DN: CN=Directory Service,CN=Windows NT,CN=Services
2. Right-click on the CN=Directory Service node and select Properties from the menu.
3. With the Attribute Editor tab selected, scroll through the list until you find the dsHeuristics attribute.
4. Double-click on the dsHeuristics attribute to open the editor. If the attribute is empty, set it with the value: 0000002. If the attribute has an existing value, make sure the seventh digit is set to 2.
5. Click OK twice.

Step 4: Enable Anonymous Access to the CRL Store of the Active Directory

Now that the ability to make anonymous connections to the directory is enabled, you must specify which attributes/entries that exist in the AD can actually be accessed via an anonymous connection.

To enable anonymous access to the CRL store of the Active Directory

1. From within the ADSI Edit console, expand the Domain node. This should display an node which contains the same name as the domain that is running on the computer (for example, DC=name,DC=com). Right-click on this node and select properties from the menu.
2. Select the Security tab and click Add. Enter anonymous in the object name field and select Check Names. This should resolve the name to ANONYMOUS LOGON. Click OK twice.
3. Select and Expand the Configuration node. Right-click on the first entry (CN=Configuration,DC=name,DC=com) and select Properties. Add the ANONYMOUS LOGON in the Security tab. This can be done using the same steps as before.
4. Expand the CN=Configuration,DC=name,DC=com node. The CRL store is located four levels under the Configuration node. You will need to add the "ANONYMOUS LOGON" user in the security tab to the following attributes:

   - CN=Services
   - CN=Public Keys Services
   - CN=CDP
   - CN=*DomainName*
   - CN=*rootCertName*

   Each attribute can be found by expanding the node listed above it. Also the last attribute is located inside the DomainName attribute.

5. You can test that the Anonymous Access has been configured correctly by using Jxplorer. Open Jxplorer and enter the following information:

   ```
   BaseDN: CN=rootCertName,CN=computerName,CN=CDP,CN=Public Key
   Services,CN=Services,CN=Configuration,DC=domainprefic,DC=com
   Host: AD computer name or IP Address
   Port: 389
   Security Level: Anonymous
   ```

6. Click OK.

   You should be able to connect to the CRL Store of the Active Directory. The first attribute is the list should be certificateRevocationList

Step 5: Configure the CRL Distribution Point for the Microsoft CA

Along with configuring the Active Directory to accept anonymous connections, you must specify the correct LDAP URL to be used as the CRLDP.

To configure the CRL distribution point for the Microsoft CA

1. Open the Certification Authority manager from Administrative Tools, right-click on the top level of the structure (it has the same name as the CN of your root certificate) and select Properties.
2. Select the Extensions tab, and make sure that CRL Distribution Point (CDP) is selected from the drop down list.

   Depending on if you have already edited this extension, the default entries listed should be:

   ```
   C:\WINDOWS\system32\CertSrv\CertEnroll\<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl
   ldap:///CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>,CN=CDP,CN=Public Key Services,CN=Services,<ConfigurationContainer><CDPObjectClass>
   http://<ServerDNSName>/CertEnroll/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl
   file://\\<ServerDNSName>\CertEnroll\<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl
   ```

3. Make sure that only the following attributes are enabled for the extensions:

   - C:\Windows…
     - Publish CRLs to this location
     - Publish Delta CRLs to this location
   - ldap:///…
     - Publish CRLs to this location
     - Publish Delta CRLs to this location
   - http://...
     - No attributes
   - File://\\..
     - No attributes

   Note: These are the default extensions. The only important one that you really need is the default LDAP URL, which allows the Microsoft CA to publish the CRL to the Active Directory. The reason the other extensions arent included in published certificates is that the Certificate authentication agent will only use the first address listed in the certificate.

   You now need to add a new extension which will be used as the CRLDP by the Certificate authentication agent.

4. Click Add, and paste the following into the Location field:

   ```
   ldap://%SERVER_DNS_NAME%/CN=%CA_NAME_HASH%%CRL_SUFFIX%,CN=%SERVER_SHORT_NAME%,CN=CDP,CN=Public Key Services,CN=Services,%CONFIG_NAME%?certificateRevocationList?base?(objectclass=cRLDistributionPoint)
   ```

   Alternatively, you can also use the following LDAP URLs:

   ```
   ldap://hostname/ipaddress:389/CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>,CN=CDP,CN=Public Key Services,CN=Services,<ConfigurationContainer>
   ```

   ```
   ldap://hostname/ipaddress:389/CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>,CN=CDP,CN=Public Key Services,CN=Services,<ConfigurationContainer>?certificateRevocationList?base?(objectClass=cRLDistributionPoint)
   ```

5. Click OK to add the new LDAP URL. Select the new LDAP CRLDP from the extension list and enable the following attributes for the new extension. Enable to following attributes:

   - Include in all CRLs
   - Include in CRLs
   - Include in the CDP extension

   Note: These should be the only attributes available for the new LDAP URL.

6. Renew the CA Certificate to include the new CRLDP location.

7. Click OK and then Yes when asked to restart the Certificate service.

   Right-click on the Revoked Certificates node in the Certification Authority, under the root certificate, and select All Tasks \ Publish. This generates a new CRL.

8. You now need to reissue the user certificates, so that they include the new CRLDP location. Issue a test certificate and check that the correct LDAP URL is included in the CRLDP extension in the certificate.

Step 6: Configure the Certificate Authentication Agent to Use CRLDP to Find Revocation Status of User Certificates

To configure the Certificate authentication agent to use CRLDP to find revocation status of user certificates, complete the following steps:

1. Open the Certificate authentication configuration file CA_certtga.ini.
2. To use the CRLDP revocation method, configure the following values:
   - AuthHostName
   - CrlDPIssuerCertPath
   - CrlDPIssuerCerts
   - CrlDPTimeOut
   - CrlPollInterval
   - RevocationMeth
   - TrustedCertNames
   - TrustedCertPath

Implement LDAP Authentication

CA SSO supports primary authentication to user stores which are LDAP compliant, such as, Microsoft Active Directory and CA Directory. This section explains how to install the LDAP authentication agent.

Name Mapping

Name mapping is the method for mapping a User Name entered in the LDAP authentication dialog to the LDAP user distinguished name (DN) in the directory. When installing the LDAP authentication agent, you can configure one of two name mapping methods:

- Substitution
- Search

The configuration settings that make up the name mapping section are:

- StaticName
- Base DN
- Filter
- Scope

During installation, one or more name mappings must be defined in sections named [NameMapping<index>], with the first index value being 0 and consequent ones being in increments of 1. If you have sections [NameMapping0] and [NameMapping7], the latter won't be read in or used.

You can update name mapping information post-installation using the LDAP authentication agent configuration file ldapPolicy.ini.

Note: While selecting the method for name mapping, consider that *search* is the preferable option for complex environments, that is, numerous levels of name mapping information, while *substitution* is the preferred option for simple one level name mapping environments.

Failover and Load Balancing (Primaries and Secondaries)

The LDAP authentication agent can distribute processing between LDAP servers. You can define two groups of LDAP servers: *Primary* and *Secondary. Primary* servers can be configured using the installation wizard, and both *Primary* and *Secondary* servers can be configured in the ldapPolicy.ini file post-installation.

The LDAP authentication agent always tries to bind to the servers from the *Primary* group first and if it fails binds to the servers defined in the *Secondary* group.

Within each group of servers, LDAP server definitions consist of two parts:

- LDAPHost<index>=<hostname>[<port number>][/bias_value]

  The first index value being 0 and consequent ones in increments of 1.

  Port number and bias value are optional, with the default values being 389 and 100 respectively. Bias value can be used to configure load-balancing between the servers within the group (i.e. an LDAP server with a bias value of 50 is half-as likely to be chosen when the decision is made as to which server to connect to).

- [<group type>.LDAPHost<index>]

  A section containing the details required for the initial administrative bind, when a connection to the server is established for the first time.

Note: At least one LDAP server must be defined in the *Primaries* group, in order for the LDAP authentication agent to initialize successfully.

The configuration parameters that make up each [<group type>.LDAPHost<index>] section are:

Name	Description
AuthenticationLevel	Specifies the level of authentication to use when connecting to the hosts LDAP directory: Anonymous, Simple or SSL
LoginName	The login name to use for Simple authentication. **Note:** Only if AuthenticationLevel= Anonymous.
Password	The password to use for Simple authentication. This value is stored in the configuration file in an obfuscated form. If configuration was done by the installer, this obfuscation will be taken care of automatically. To alter this value manually, use the 'ssoencconf.exe' tool supplied with the authentication agent. **Note:** Only if AuthenticationLevel= Anonymous.
Keystore	The name and path of a PEM file key store to be used when SSL authentication is used. **Note:** Only if AuthenticationLevel=SSL

If any of the parameters required for non-*Anonymous* authentication are missing, the LDAP authentication agent fails to start. If any of the parameters required for SSL communication are missing, the LDAP authentication agent fails to start.

LDAP Authentication Level

The LDAP authentication level defines the authentication type used for binding to the LDAP directory, for example, Active Directory or CA Directory. When installing the LDAP authentication agent, you can configure one of three communication options:

- Anonymous
- Simple
- Secure Socket Layer (SSL)

You can update the authentication level post-installation using the LDAP authentication agent configuration file ldapPolicy.ini.

Anonymous Authentication

Anonymous authentication allows users to connect to an LDAP directory without providing a username and password.

Simple Authentication

Simple authentication allows users to connect to a directory by providing a username and password. The following conditions are required for the bind to be considered Simple:

- The name corresponds to a real entry in the directory.
- That entry has a password attribute.
- The supplied password matches it.

To implement simple authentication, provide the following information during the LDAP authentication agent installation:

User DN

Defines the fully qualified distinguished name (DN) of a user with administrative privileges defined on the LDAP server.

Password

Defines the password for the user whose DN was specified using the UserDN configuration.

SSL Authentication

Strong authentication uses SSL certificates to protect LDAP access by encrypting data with Secure Sockets Layer (SSL) security. When certificate based authentication is used, all communication on the association set up by the connection uses SSL encryption.

SSL certificate based authentication is typically used in environments where personal or company data requires protection.

To implement SSL authentication, provide the following information during the LDAP authentication agent installation:

Keystore

Defines a full path to the store containing the Agent certificate for SSL communication.

Note: Use the ..\certs\CAcert.pem file created in the section, Configure SSL Communication between the CA SSO Server and the CA Directory and Active Directory Data Store (Windows) (see page 226) to configure Enable Ldap Over SSL communication between the LDAP Authentication Host and Active Directory

Before You Install

This section describes what you need to know or do before you install the LDAP authentication agent. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Create Users in an LDAP User Data Store

This procedure guides you through creating two users in an LDAP user data store. You can use this information during the LDAP authentication agent installation process to test authentication with the CA SSO Server. This example uses the ps-ldap data store (CA Directory) that is created by default during the CA SSO Server installation.

Note: This procedure requires both the CA SSO Server and Policy Manager to be installed, and only works with CA Directory.

To create users in an ldap user data store

1. Open the Policy Manager.
2. Click the Users icon and navigate to the ps-ldap datastore.
3. Create two new users in the ps-ldap data store.

   Admin

   This user configures the LDAP authentication agent.

   LDAPuser

   This user account tests the LDAP authentication method.

4. For both users, assign the LDAP authentication method, and set a password for the LDAP authentication method.
5. Click Resources, Single Sign-On Resources, User Resources, Datastores, and right-click the ps-ldap user data store and select Properties.
6. Note the following properties of the ps-ldap datastore:
   - Base Path
   - Port Number

Configure the CA SSO Client for LDAP Authentication

You can use the Auth.ini file to configure the user's LDAP authentication settings, including:

- Server name and port information
- Authentication method

To configure the CA SSO Client for LDAP authentication

1. Open the Auth.ini file.
2. Add LDAP to the *AuthMethods* setting, for example:

   ```
   [ServerSet1]
   Name=ServerSet1
   PolicyServers=ssops-a
   AuthMethods=LDAP CERT SSO
   ```

3. Add the LDAP server name to the *AuthLDAP* setting. For example:

   ```
   AuthLDAP=server1100
   ```

4. (Optional) If the default LDAP port number is already in use, configure an alternative LDAP server port number. For example:

   ```
   AuthLDAP=server1100:17979
   ```

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the authentication agent:

- Ensure that all system requirements are met before you install the authentication agent.

  For more information, see the *SSO readme* file.

- Ensure that the CA SSO Server has been installed.
- Ensure that the computer you are installing the agent on has TCP/IP communications with the CA SSO Client computers.

- Ensure you know all relevant information prior to running the installation including:
  - The number of LDAP authentication agents to be installed.
  - The host name and port number of each LDAP authentication agent machine.
  - The initial authentication connection details, including:
    - Admin user name and password for Simple connection
    - Keystore for an SSL connection
  - The number of name mappings for each authentication agent and choice of Search or Substitution name mapping methods.
  - The authentication host and encryption key details from the Policy Manager.
- For silent installs, you must first run the LDAP authentication agent wizard and read the license agreement. The command line option for accepting the license agreement can be found at the bottom of the license agreement text.
- If you want to test the LDAP connection during the installation process, ensure you have created users in an LDAP user data store.
- If you are using the LDAP authentication agent with Active Directory, ensure that you install the LDAP authentication agent with a base DN other than the Active Directory root. For example: DN=”cn=Finance,dc=MyDomain,dc=com”. You can add multiple base DNs.

  Note: If the base DN is configured to the Active Directory root, the LDAP authentication agent may hang when a user attempts to log onto the CA SSO Client.
- Ensure all operating system clocks produce a reliable and correct timestamp for the time-zone where each computer hosting any CA SSO components is located.

More information:

Create Users in an LDAP User Data Store (see page 272)
Synchronize Operating Systems (see page 246)

Install the LDAP Authentication Agent

This section explains how to install the LDAP authentication agent.

Install Using the Wizard

This topic explains how to install the LDAP authentication agent using the Product Explorer.

To install the LDAP authentication agent using the wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer wizard expand the CA SSO Authentication Agents folder, and select LDAP Authentication Agent.

   The Install button becomes active.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install Using Silent Installation

You can install the LDAP authentication agent silently. This means that you need to provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the LDAP authentication agent at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select LDAP Authentication Agent.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the LDAP authentication agent using silent installation.

4. Open a command prompt and navigate to the LDAP authentication agent folder on the product DVD.

5. From the command prompt, type:

   setup.exe -silent -V LICENSE_VIEWED=*value* {*parameters*}

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

Setup Command—Install LDAP Authentication Agent

The command line parameters for installing the LDAP authentication agent include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

```
-V IS_REBOOT_NOW=[Value]
```

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V AuthHostNameValue

Defines the authentication host in the SSO Server for this agent type.

The command has the following format:

```
-V AuthHostNameValue=[Value]
```

Value: The name of the authentication host.

-V AuthenticationLevelValue

Defines the authentication type used for binding to the LDAP server.

The command has the following format:

```
-V AuthenticationLevelValue=[Value]
```

Value: The type of authentication, for example, Anonymous, Simple or SSL.

-V KeystoreLocation

Defines the full path for the store containing the agent certificate for SSL communication.

The command has the following format:

```
-V KeystoreLocation=[Value]
```

Value: The full path to the location of the keystore. Surrounded by quotation marks if the path contains spaces.

-V LDAPHostNameValue

Defines the directory computer name.

The command has the following format:

```
-V LDAPHostNameValue=[Value]
```

Value: The name of the computer hosting the LDAP directory.

-V LDAPPortValue

Specifies the port number of the LDAP directory.

The command has the following format:

```
-V LDAPPortValue=[Value]
```

Value: The port number of the LDAP directory.

-V LDAPUserDNValue

Define the logon name for the initial bind.

The command has the following format:

```
-V LDAPUserDNValue=[Value]
```

Value: The name used to log on for the initial bind.

-V LDAPUserPasswordValue

Defines the logon password for the initial bind.

The command has the following format:

```
-V LDAPUserPassword=[Value]
```

Value: The password used for the initial bind.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

```
-V LICENSE_VIEWED=[Value]
```

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V SearchBaseDnValue

Specifies the base DN for the LDAP search.

The command has the following format:

-V SearchBaseDnValue=[*Value)*

Value: The base DN for the LDAP search.

-V SearchFilterValue

Specifies a filter for the LDAP search.

The command has the following format:

-V SearchFilterValue=[*Value*]

Value: The filter for the LDAP search.

-V SearchScopeTypeValue

Specifies the scope for the LDAP search if the NameMapping type Search is used.

The command has the following format:

-V SearchScopeTypeValue=[*Value]*

Value: The scope for the LDAP search if the NameMapping type Search is used.

-V StaticNameValue

Defines the DN format of the user repository if the NameMapping type Substitution is used.

The command has the following format:

-V StaticNameValue=[*name*]

Value: The DN format of the user repository, for example cn=%s ou=users.

-V TicketEncryptionKeyValue

Specifies the encryption key for the authentication host.

The command has the following format:

-V TicketEncryptionKeyValue=[*Value*]

Value: The encryption key for the authentication host.

-V NameMappingTypeValue

The command has the following format:

-V NameMappingTypeValue=[*Value*]

-V IsActiveDirectoryAware

The command has the following format:

-V IsActiveDirectoryAware=[*Value*]

-V DisplayFailureMessageDetail

The command has the following format:

-V DisplayFailureMessageDetail=[*Value*]

-V AGENT_ID_FILE

Defines the Trusted Certificate file in .pem format.

The command has the following format:

-V AGENT_ID_FILE=[FilePath]

FilePath: Indicates the absolute path to the file (along with file name)

-V AGENT_KEY_FILE

Defines the private key file in .pem format.

The command has the following format:

-V AGENT_KEY_FILE =[FilePath]

FilePath: Indicates the absolute path to the file (along with file name)

-V COMM_MODE

Defines the mode of communication.

The command has the following format:

-V COMM_MODE=[Value]

Value: 0 | 1 | 2

- 0 - Non-FIPS mode
- 1 - FIPS-only
- 2 - Mixed mode

Install Using Silent Installation and Response File

Use the following procedures to install the LDAP authentication agent silently using a response file.

Note: You can use a combination of response file and command line options to silently install the LDAP authentication agent. In this case, the command line options will override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the LDAP authentication agent folder on the product DVD.
3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create an LDAP Authentication Agent Response File (see page 281)
Setup Command—Install LDAP Authentication Agent (see page 277)

Create an LDAP Authentication Agent Response File

Response files contain user specified install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the LDAP authentication agent folder on the product DVD.

2. From the command prompt, enter:

```
setup.exe -options-record {file name}
```

-options-record file name

Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   A response file is created in the directory specified.

Post-Installation Configuration Options

The following sections explain some of the post-installation configuration options.

Configure ldapPolicy.ini and CA_ldaptga.ini Settings

Once you have installed the authentication agent, you can configure the ldapPolicy.ini and CA_ldaptga.ini settings files with any post-installation changes.

LDAP Authentication and Active Directory Integration

You can set up LDAP authentication to allow authentication against an Active Directory.

In addition to generic LDAP functionality, Active Directory specific options have been added to the LDAP authentication method providing support for a number of specific password and password policy features.

To use these features, the target directory must be an Active Directory, and the relevant configuration values must be set in the [LDAPConnection] section of the CA_ldaptga.ini file. For more information on these values, refer to the *IsActiveDirectoryAware* and *DisplayFailureMessageDetail* values in Configuring Authentication Agents in the *CA SSO Administration Guide*.

Note: All features are reliant on the relevant Active Directory's settings. For example, if the user's password is configured in Active Directory to never expire, the user will never be presented with a password expiry notification.

Active Directory features include:

- Password change
- Notification and handling of password expiry
- Notification of the reason for authentication/password change failure

Password Change

When using the Active Directory extension, the LDAP authentication agent allows the user to change their password, for example, by using the *Change Authentication Details* button on the Details page of the SSO Tools application.

Notification and Handling of Password Expiry

If the user's password has an upcoming expiry date the user is warned of the password expiry as per normal Windows authentication. That is, the relevant registry setting is queried to determine whether to notify the user of the number of days to password expiry. The user has the ability to update their password in response to the notification.

Note: If authentication fails due to password expiry, the user is notified and provided with an opportunity to change their expired password.

Notification of the Reason for Authentication/Password Change Failure

The user is given details of the reason for an authentication failure; in particular the user can distinguish between the following types of authentication failure:

- Username/password entered does not match known credentials
- Account is locked/disabled
- Attempt to logon outside of permissible hours
- Password has expired
- Password history prevents re-use of the specified password
- Password minimum age prevents password change
- New password is not of the required complexity (and why)

Implement RSA Authentication

CA SSO supports primary authentication using RSA SecurID. This section explains how to implement RSA SecurID authentication.

Before You Install

This section describes what you need to know before you install the RSA authentication agent. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Register the Authentication Host as an Agent Host

In addition to installing RSA SecurID, you must complete the following steps to ensure the RSA SecurID authentication agent can communicate with the ACE Server.

1. Your ACE Server administrator must register the server that the RSA authentication agent is installed on as an Agent Host.
2. You must copy the sdconf.rec file to the Windows system32 folder on the machine where you have installed the RSA SecurID authentication agent.

   For more information, contact your ACE Server administrator.

   Note: The user in the CA SSO Server must have the same logon name as on the RSA ACE Server.

   There must be a TCP/IP connection between the ACE Server and RSA authentication agent machine.

setup Command—Install RSA Authentication Agent

The command line parameters for installing the RSA SecurID authentication agent include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-V AuthHostNameValue

Specifies the authentication host in the SSO Server for this agent type.

The command has the following format:

```
-V AuthhostNameValue=[Value]
```

Value: The name of the authentication host.

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

```
-V IS_REBOOT_NOW=[Value]
```

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

```
-V LICENSE_VIEWED=[Value]
```

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V TicketEncryptionKeyValue

Specifies the encryption key for the authentication host.

The command has the following format:

```
-V TicketEncryptionKeyValue=[Value]
```

Value: The encryption key for the authentication host.

-V AGENT_ID_FILE

Defines the Trusted Certificate file in .pem format.

The command has the following format:

```
-V AGENT_ID_FILE=[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

-V AGENT_KEY_FILE

Defines the private key file in .pem format.

The command has the following format:

```
-V AGENT_KEY_FILE =[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

-V COMM_MODE

Defines the mode of communication.

The command has the following format:

```
-V COMM_MODE=[Value]
```

Value: 0 | 1 | 2

- 0 - Non-FIPS
- 1 - FIPS-only
- 2 - Mixed mode

Configure the CA SSO Client for SecurID Authentication

You can use the Auth.ini file to configure the user's RSA SecurID authentication settings on the CA SSO Client side, including:

- Server name
- Authentication method

To configure the CA SSO Client for SecurID authentication

1. Edit the Auth.ini file to include RSA as one of the authentication methods, for example:

   ```
   [ServerSet1]
   AuthMethods=RSA
   ```

2. Edit the Auth.ini to include the name of the RSA authentication server. For example:

   ```
   [ServerSet1]
   AuthRSA=server1
   ```

3. (Optional) Configure the RSA server port number. For example:

   ```
   AuthRSA=server1:19987
   ```

   If the port number is not specified, the default port (13987) is used.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the authentication agent:

- Ensure that all system requirements are met before you install the authentication agent.

  For more information, see the *SSO readme* file.

- Ensure that the CA SSO Server has been installed.
- Ensure that the computer you are installing the agent on has TCP/IP communications with the CA SSO Client computers.
- Ensure you know all relevant information prior to running the installation including:
  - The name of authentication agent machine
  - The RSA authentication agent encryption key
- For silent installs, you must first run the RSA SecurID authentication agent wizard and read the license agreement. The command line option for accepting the license agreement can be found at the bottom of the license agreement text.
- Ensure all operating system clocks produce a reliable and correct timestamp for the time-zone where each computer hosting any SSO components is located.

More information:

Synchronize Operating Systems (see page 246)

Install the RSA SecurID Authentication Agent on Windows

This section explains how to install the RSA SecurID authentication agent.

Install Using the Wizard

This topic explains how to install the RSA SecurID authentication agent using the Product Explorer.

To install using the wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer wizard, expand the CA SSO Authentication Agents folder, and select RSA Authentication Agent.

   The Install button becomes active.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist.

Install Using Silent Installation

You can install the RSA SecurID authentication agent silently. This means that you need to provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the RSA authentication agent at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer, expand the CA SSO Authentication Agents folder, and select RSA SecurID Authentication Agent.

   The Install button becomes active.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the RSA SecurID authentication agent using silent installation.

4. Open a command prompt and navigate to the RSA SecurID authentication agent folder on the product DVD.

5. From the command prompt, type:

   ```
   setup.exe -silent -V LICENSE_VIEWED=value {parameters}
   ```

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

Install Using Silent Installation and Response File

Use the following procedures to install the RSA SecurID authentication agent silently using a response file.

Note: You can use a combination of response file and command line options to silently install the RSA SecurID authentication agent. In this case, the command line options override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the RSA SecurID authentication agent folder on the product DVD.
3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create an RSA SecurID Response File (see page 291)
setup Command—Install RSA Authentication Agent (see page 285)

Create an RSA SecurID Response File

Response files record user specific install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the RSA SecurID authentication agent folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record {file name}
   ```

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.
3. Complete the installation.

   A response file is created in the directory specified.

Post-Installation Configuration Options

This section explains some of the configuration options you can implement post-installation.

Configure CA_rsatga.ini Settings

Once you have installed the authentication agent, you can configure the CA_rsatga.ini settings file with any post-installation changes.

Re-install the RSA SecurID Authentication Agent

If you uninstall the RSA SecurID authentication agent and then re-install it, you must alter the configuration on the RSA ACE server:

1. Open the Edit Agent Host dialog.
2. De-select the Sent Secret Node check box.

Implement Windows Authentication

CA SSO supports user authentication to Windows, using Active Directory as the authentication provider.

Before You Install

This section describes what you need to know or do before you install the Windows authentication agent. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Set Up a Domain Controller

Before you start installing the Windows authentication agent, make sure to set up your domain controller. You also need to ensure that all machines hosting the CA SSO Client, CA SSO Server, and the Windows authentication agent are connected to the network.

Create an Authentication Host Entry on the CA SSO Server

You only need to do this if you decide to use a different authentication host to the default created during the CA SSO Server installation (WIN_Authhost).

Create Users

For CA SSO users to be able to use the Windows authentication method, each must be created with the same name as a user on the domain controller. They must also be given access to the authentication host you are using for Windows authentication. For example, a user named fred would be fred.picard.net and have access to WIN_Authhost and the WIN authentication method.

Create a User Alias

Enter the following command into a selang session to change the authentication host and user values for the ones you have used:

```
er authhost <auth_host_name> useralias(“<user>=<alias>”)
```

SSL Communication

The use of SSL is mandatory for the Windows authentication agent. To set this up during installation, you must specify:

- An Identity file and password. You can also specify a Trust file (optional).
- To install the TGA, an administrator will require (at least) a 'P12' file containing certificate(s) and associated private key that can be used by the server to assert its identity.
- To install the client components, an administrator will require each client to have (at least) a pem file containing the required trusted certificates with which the client can confirm the servers identity.

  Note: CA SSO does not provide the tools/utilities to create these files. You can choose to use your PKI design and technology adoption, or download the OpenSSL tool which guides you through the trusted certificate creation process. For more information, see the procedures at the end of this section.

Identity File and Password

The identity file is a PKCS #12 (Personal Information Exchange Syntax Standard) format file containing the private key and machine certificate of the authentication host. This is required to authenticate SSL communication between the authentication host and CA SSO Client machines.

More information:

4. Create an Identity File (PKCS#12) for the Windows Authentication Agent (see page 298)

Trust File

The Trust file is the PEM format issuer certificate of the identity files installed on the CA SSO Client machines. This is required if the SSL communications between the CA SSO Client and Windows authentication agent are to be bilaterally authenticated.

More information:

Create a Self Signed Certificate (see page 294)
Issue a Certificate for the Windows Authentication Agent (see page 296)

Create a Self Signed Certificate

Creation of self signed trust certificates vary depending on your PKI design and technology adoption. The following process takes you through creating a self signed certificate using OpenSSL.

To create a self signed certificate

1. Download and install OpenSSL.
2. Generate RSA keypair for the CA.
3. Generate the CA cert request.
4. Create the CA certificate.

1. Download and Install OpenSSL

To download and install OpenSSL

1. Download the latest version of openSSL from http://www.openssl.org.

2. Generate the RSA Keypair for the Certificate Authority

The following procedure takes you through creating the keypair, which includes the creation of both a public and private key.

To generate the RSA keypair, open a command prompt and type:

```
openssl genrsa -out [ca_keyfile.pem] [keylength]
```

ca_keyfile.pem

Defines the name of the keypair file.

keylength

Defines the size of the key in bits, for example 2048. 2048 or higher is recommended for RSA keys for security reasons.

Note: If you want to add password protection, include the command *-des3,* for example:

```
openssl genrsa -des3 -out [ca_keyfile.pem] [keylength]
```

3. Generate the CA Certificate Request

This procedure takes you through creating the certificate request. The certificate request can then be self signed or sent to a certificate authority for an official signature.

To generate the CA certificate request, open a command prompt and type:

```
openssl req -new -key [ca_keyfile.pem] -keyform PEM -out [ca_cert_request.pem] -outform PEM -subj
[ca_subject_name_dn]
```

ca_keyfile.pem

Defines the name of the keypair file.

ca_cert_request.pem

Defines the name of the Certificate Authority request file.

ca_subject_name_dn

Defines the distinguished name, for example:

"/C=AU/O=Sample Organization/OU=Samples/CN=Sample CA"

4. Create the Certificate Authority File

This procedure takes you through self signing the certificate.

To create the Certificate Authority file, open a command prompt and type:

```
openssl x509 -inform PEM -outform PEM -req -in [ca_cert_request.pem] -signkey [ca_keyfile.pem] -keyform PEM
-out [ca_cert.pem] -extfile [path_to_openssl.cnf] -extensions [extension_in_openssl.cnf] -days [days_valid]
```

ca_cert_request.pem

Defines the name of the Certificate Authority request file.

ca_keyfile.pem

Defines the name of the keypair file.

ca_cert.pem

Specifies the name of the generated Certificate Authority file.

path_to_openssl.cnf

Defines the path to the openssl.cnf file.

extension_in_openssl.cnf

Identifies which extension section to use in the openssl.cnf file, for example v3_ca.

days_valid

The number of days the certificate is valid.

Issue a Certificate for the Windows Authentication Agent

The following procedure takes you through issuing a certificate for the Windows authentication agent signed by a Certificate Authority.

To issue a certificate for the Windows authentication agent

1. Generate RSA keypair.
2. Generate the agent certificate request.
3. Issue the agent certificate.
4. Create an identity file (PKCS#12) for Windows authentication agent.

1. Generate the RSA Keypair for the Windows Authentication Agent

The following procedure takes you through creating the keypair, which includes the creation of both a public and private key.

To generate the RSA keypair for the Windows authentication agent, open a command prompt and type:

```
openssl genrsa -out [agent_keyfile.pem] [agent_keylength]
```

agent_keyfile.pem

Defines the name of the Windows authentication agent keypair file.

agent_keylength

Defines the size of the key in bits, for example 2048. 2048 or higher is recommended for RSA keys for security reasons.

2. Generate the Agent Certificate Request

This procedure takes you through creating the certificate request. The certificate request can then be self signed or sent to a certificate authority for an official signature.

To generate the agent certificate request, open a command prompt and type:

```
openssl req -new -key [agent_keyfile.pem] -keyform PEM -out [agent_cert_request.pem] -outform PEM -subj
[agent_subject_name_dn]
```

agent_keyfile.pem

Defines the name of the keypair file.

agent_cert_request.pem

Defines the name of the Certificate Authority request file.

agent_subject_name_dn

Defines the distinguished name, for example:

```
"/C=AU/O=Sample Organization/OU=Samples/CN=Win Auth Agent"
```

3. Issue the Agent Certificate

This procedure takes you through self signing the certificate.

To issue the agent certificate, open a command prompt and type:

openssl x509 -inform PEM -in [*agent_cert_request.pem*] -req -days [*days_valid*] -CA *[ca_cert.pem*] -CAkey [ca_*keyfile.pem*]   -set_serial [*agent_cert_serial_number*] -extfile [*path_to_openssl.cnf*] -extensions [*extension_in_openssl.cnf*] -out *[agent_cert.pem*]

agent_cert_request.pem

Defines the name of the Certificate Authority request file.

days_valid

The number of days the certificate is valid.

ca_cert.pem

Specifies the name of the generated Certificate Authority file.

ca_keyfile.pem

Defines the name of the keypair file.

agent_cert_serial_number

The authentication agent's certificate serial number. Can be any number greater than 0.

path_to_openssl.cnf

Defines the path to the openssl.cnf file.

extension_in_openssl.cnf

Identifies which extension section to use in the default openssl.cnf file, for example v3_req.

agent_cert.pem

Specifies the name of the generated authentication agent certificate.

4. Create an Identity File (PKCS#12) for the Windows Authentication Agent

The output agent_identity.p12 file is used as the Identity File for the Windows authentication agent. The path to this file is configured for the IdentityPassword variable in the CA_wintga.ini file

The password entered when this command is run must be configured for the IdentityPassword variable.

Note: The password set is not the cleartext password but the obfuscated string output when running the *ssoencconf* tool. This is in the bin dir and is run by: *ssoencconf -d [password]*.

To create an identity file (PKCS#12) for the Windows authentication agent, open a command prompt and type:

```
#openssl pkcs12 -name [PKCS#12_friendly_name] -in [agent_cert.pem] -inkey [agent_keyfile.pem] -out
[agent_identity.p12] -export
```

PKCS#12_friendly_name

Specifies the friendly name of the PKCS#12 file. This name can be anything, for example, Win Auth Agent.

agent_cert.pem

Specifies the name of the authentication agent's Certificate Authority file.

agent_keyfile.pem

Defines the name of the keypair file.

agent_identity.p12

Specifies the name of the generated Identity File.

Configure the CA SSO Client for Windows Authentication

You can use the Auth.ini file to configure the user's Windows authentication settings, including:

- Server name
- Authentication method

To configure the CA SSO Client for Windows authentication

1. Edit the Auth.ini file to include WIN as one of the authentication methods, for example:

   ```
   [ServerSet1]
    AuthMethods=WIN
   ```

2. Edit Auth.ini to include the name of the Windows authentication host in the auth agent keyname. For example:

   ```
   [Serverset1]
    AuthWIN=server1
   ```

3. Edit the Auth.ini to specify which domain controller to use. For example:

   ```
   [auth.WIN]
    NearestDomainController=Yes
   ```

4. Specify the values of the other settings associated with WIN authentication in the [Auth.WIN] section of the Auth.ini file. For example:

   ```
   ConnectionTimeout=
   IdentityFile=C:\Certs\myIdentityFile.p12
   IdentityPassword=<obfuscated password>
   TrustFile= C:\Certs\myTrustFile.pem
   AutoNetworkAuth=0
   NearestDomainController=0
   ```

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the authentication agent:

- Ensure that all system requirements are met before you install the authentication agent.

  For more information, see the *SSO readme* file.

- Ensure that your CA SSO Servers have been installed and configured.

- Ensure you know all relevant information before running the installation including:
  - The trust file (.pem). Used to verify the identity of the WinTicketAgent.
  - Identity file (.p12) and password. Used by the WinTicketAgent to establish its identity to the WinTicketInterface.
  - The name of the authentication host.
  - Authentication host encryption key.

    By default, the CA SSO Server installation creates a WIN_Authhost entry with a randomly-generated key value.
- Ensure all operating system clocks produce a reliable and correct timestamp for the time-zone where each computer hosting any CA SSO components is located.

More information:

Synchronize Operating Systems (see page 246)
Trust File (see page 293)
Identity File and Password (see page 293)

Install the Windows Authentication Agent

This section explains how to install the Windows authentication agent.

Authentication Agents Backward Compatibility with CA SSO Clients

r12 Authentication Agents support backward compatibility with r8.1 Clients.

Install Using the Wizard

This topic explains how to install the Windows authentication agent using the Product Explorer.

To install using the wizard

1. Log onto the computer on the network with administrative rights where you intend to install the authentication agent.
2. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

3. From the Product Explorer*,* expand the CA SSO Authentication Agents folder, and select Windows Authentication Agent.

   The Install button becomes active.

4. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist.

Install Using Silent Installation

You can install the Windows authentication agent silently. This means that you need to provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the Windows authentication agent at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Windows Authentication Agent.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the Windows authentication agent using silent installation.

4. Open a command prompt and navigate to the Windows authentication agent folder on the product DVD.

5. From the command prompt, type:

```
setup.exe -silent -V LICENSE_VIEWED=value {parameters}
```

-silent

Specifies a silent install.

-V LICENSE_VIEWED=value

Specifies whether you have viewed the license agreement found in the product install wizard.

parameters

Specifies the options to include in the silent install.

For more information on command line options, see the next topic.

setup Command—Install Windows Authentication Agent

The command line parameters for installing the Windows authentication agent include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-V AuthHostNameValue

Defines the authentication host in the SSO Server for this agent type.

The command has the following format:

```
-V AuthHostNameValue=[Value]
```

Value: Name of the authentication host.

-V TicketEncryptionKeyValue

Specifies the encryption key for the authentication host.

The command has the following format:

```
-V TicketEncryptionKeyValue=[Value]
```

Value: The encryption key for the authentication host.

-V IdentityFileLocation

Defines the location for the identity file.

The command has the following format:

```
-V IdentifyFileLocation=[Value]
```

Value: Location of the identity file.

-V IdentityPasswordValue

Specifies the password for the identity file.

The command has the following format:

-V IdentityPasswordValue=[*Value*]

Value: The password for the identity file.

-V TrustFileLocation

Defines the location of the trust file.

The command has the following format:

-V TrustFileLocation=[*Value*]

Value: Location of trust file.

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

-V IS_REBOOT_NOW=[*Value*]

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

-V LICENSE_VIEWED=[*Value*]

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V COMM_MODE

Defines the mode of communication.

The command has the following format:

-V COMM_MODE=[Value]

Value: 0 | 1 | 2

- 0 - Non-FIPS mode
- 1 - FIPS-only mode
- 2 - Mixed mode

-V FIPSIdentityFile

Defines the Trusted Certificate file in .pem format.

The command has the following format:

```
-V FIPSIdentityFile=[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

-V FIPSPrivateKeyFile

Defines the private key file in .pem format.

The command has the following format:

```
-V FIPSPrivateKeyFile =[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

-V FIPSTrustedCertFile

Defines the Agent Trust file. It can be in the following formats based on the communication mode configured.

The command has the following format:

In case of FIPS only mode, it should be in .pem file format.

In case of Mixed mode, it should be either:

- .pem file format or
- .p12 file format which is generated using FIPS supported algorithms

```
-V FIPSTrustedCertFile=[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

Install Using Silent Installation and Response File

Use the following procedures to install the Windows authentication agent silently using a response file.

Note: You can use a combination of response file and command line options to silently install the Windows authentication agent. In this case, the command line options override the response file. However, we recommended you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the Windows authentication agent folder on the product DVD.

3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create a Windows Authentication Agent Response File (see page 306)
setup Command—Install Windows Authentication Agent (see page 303)

Create a Windows Authentication Agent Response File

Response files contain user specified install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the Windows authentication agent folder on the product DVD.
2. From the command prompt, enter:

   setup.exe -options-record {*file name*}

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   A response file is created in the directory specified.

Post-Installation Configuration Options

The following sections explain some of the post-installation configuration options.

Configure CA_wintga.ini Settings

Once you have installed the authentication agent, you can configure the CA_wintga.ini settings file with any post-installation changes.

Creating a Custom Authentication Agent

CA SSO offers a number of out-of-the-box methods for primary authentication, for example, SSO, Windows, LDAP and RSA SecurID as discussed in this chapter. However, you may want to develop your own specific authentication mechanism, for example, a biometric provider might want to integrate their solution into CA SSO.

CA SSO provides the functionality to accomplish this by giving you the tools to develop code that integrates with a defined SSO code interface. This interface is defined and the information is supplemented using a sample integration MS VC++ project that can be requested from your CA representative. This sample demonstrates the steps you must follow to develop your own authentication agent that integrates with CA SSO.

Program Architecture

All CA SSO authentication agents have a similar architecture. Each authentication agent has three components:

- A graphical user interface (GUI) – resides on the CA SSO Client
- An open authentication engine (OAE) – resides on the CA SSO Client
- A ticket-granting agent (TGA) – resides on an SSO Authentication Host

The GUI Component

The GUI DLL provides the CA SSO Client with an Authentication dialog, which is defined by the interface function authenticate_Dlg.

The OAE Component

The open authentication engine is also known as the interface library.

This library provides the CA SSO Client with an interface for requesting authentication defined by the oae_GetTicket function.

The OAE also provides a call-back function for the GUI component defined by the AuthCb_Verify function. This function is triggered when the OK button is pressed on the Login dialog. The OAE then sends a TCP/IP request to the TGA component. In this way this part of the authentication agent is responsible for communication between the GUI and the TGA.

The TGA Component

This agent communicates directly with the authentication server. It also communicates with client-side library components through TCP/IP.

The Windows version has the same architecture. However, there are differences between the tools that each operating system uses to create functions such as sub-processes, threads and inter-process communication.

The encrypted TCP/IP communication between authentication agent components is implemented using core tcpxdr and tcpcomm components. Logging is done using log4cpp.

Chapter 11: Adding Applications

This chapter describes how to add applications to the CA SSO system so that you can allocate them to users.

CA SSO automates the process of end-users logging onto the applications. Before end-users can start using CA SSO, a set of logon scripts have to be written. You need a logon script for every application that users need to access from CA SSO.

The logon script is a sequence of instructions that automate the logon process. The primary task of the logon script is to simulate users' actions when they log into an application and insert their user credentials (user name and password, for example) when required. Additionally, a logon script may contain procedures for other tasks associated with the logon process, such as changing a password and letting the CA SSO Server know the outcome of the logon attempt.

For more information about adding applications to CA SSO, see Launching Applications in the *CA SSO Administration Guide.*

This section contains the following topics:

How Logon Scripts Work

Whenever an authorized user selects a CA SSO supported application, the CA SSO interpreter receives the logon script and the logon data from the CA SSO Server and executes the script.

A logon script needs to conform exactly to the specific logon requirements of an application, mimicking the data entry and actions of an end-user of that application in your system. Therefore, the person writing CA SSO logon scripts needs to work together with an applications administrator who has a detailed knowledge of the logon process for each application.

These logon scripts are written in an extended version of Tcl, a scripting language that gives you the use of variables, conditions, loops, procedures, and other common programming constructs with a minimum of complexity. Prior experience with Tcl is not required, but the scriptwriter should be familiar with the applications involved and, in particular, the logon processes. For a full description of the SSO scripting language and writing logon scripts, see the *CA SSO Tcl Scripting Reference Guide*.

The SSO Interpreter is a CA SSO component that executes the Tcl scripts. Once the SSO Interpreter has carried out all the procedures in the logon script, the application continues to run with no further input from CA SSO.

To enable application-specific logon scripts to serve various users, CA SSO maintains separate logon variables for each authorized user for each application. The logon scripts refer to these logon variables for individual logon name and password and other data that may be necessary.

Decide What You Want the Script to Do

Begin by deciding what you want the script to do. A simple example might be that you want to launch an application, enter the user credentials and click OK.

You can also create scripts that perform quite complicated logons, or scripts that automate part of the process and require user input before they progress any further.

Document the Process That You Want to Automate

You must run through the process manually and document every step. This is what the script automates.

For example:

1. Launch the application.
2. Wait for the logon box.
3. Enter the username.
4. Tab to the next field.
5. Enter the password.
6. Click OK.

Make sure that you understand all the possible variables that might occur, for example, whether users are periodically prompted to change their password. The script that you write must be able to handle these exceptions.

Identify Where the Data is Stored

Before you start you must know where the following information is stored:

- The application executable location (you need to know the location of the application for which you are writing a script so that you can add this information into the SSO script)
- Logon script
- User data store

Developing Logon Scripts

The security or system administrator in charge of CA SSO is usually responsible for preparing the logon scripts. Generally, programmers write logon scripts under the administrator’s supervision.

Following is an example of the main portion of a logon script for a telnet client that comes with Windows NT:

```
# run the NT telnet client
sso run -path telnet.exe

# connect to the remote host
sso menu -item “Connect/Remote System”
sso setfield -label “Host Name” -value $_HOST
sso click -label Connect

# verify that the telnet window appears
sso window -title Telnet

# wait for the user ID; respond
sso waittext -text “logon:”
sso type -text “$_LOGINNAME{enter}”

# wait for the password prompt; respond
sso waittext -text “password:”
sso type -text “$_PASSWORD{enter}”

# wait for the system prompt
sso waittext -text “>“
...
```

The logon variables that appear in this logon script are $_HOST, $_LOGINNAME, and $_PASSWORD. The SSO Interpreter on the user’s workstation replaces these variables with the values received from the CA SSO Server.

Symbol	**Meaning**
$	Tcl variables
$_	SSO logon variables
#	Comment

For a full explanation of logon scripts, see the CA SSO *Tcl Scripting* Reference *Guide*.

You can also use the SSO Application Wizard to add your applications to SSO without the need to learn Tcl scripting for basic scripting tasks.

Note: Ensure that the Tcl SSO script is less than 262144 bytes. If any SSO script exceeds this amount you should either try to adjust the script size, or you should change the SSO Client ReceiveBufferSize (using a text editor to modify this setting in the Client.ini file) and the CA SSO Server SendBuffSize (using the Policy Manager to modify this setting in Resources, Configuration Resources, Policy Server Settings, Communication).

More information:

Using the SSO Application Wizard to Create SSO Scripts (see page 320)

Logon Variables

The logon variables include the logon script and the logon data sent to the SSO Client. These variables are fetched from the data stores. Some variables pertain to the current application, some are specific to the current user in relation to the current application, and some may hold installation-wide data.

The logon variables are stored in the LDAP and CA Access Control data store in the user’s record as properties of the LOGONINFO section. Some of the logon variables are used for authentication (*logon credentials*) and others provide operational and auditing information (such as time of last logon).

The logon variables are stored as properties of the LOGININFO section of the user's record in either the LDAP data store (for normal users) or the CA Access Control data store (for administrative users). Some of the logon variables are used for authentication (logon credentials) and others provide operational and auditing information (such as time of last logon).

Learn Mode (First Logon Situation)

To reduce the amount of configuration needed, CA SSO has a *learn mode* that functions during the first logon to an application and lets the end user provide the logon credentials for the application.

If the user credentials needed for an application are not found in the user record, the CA SSO Server and SSO Client assume that this is the first time the user is logging into the application using CA SSO. CA SSO then enters learn mode (also called the *first logon situation*), as follows:

1. The CA SSO Server notifies the SSO Client that no credentials are available.
2. The SSO Client displays a Set Login Information dialog box that prompts the user for user credentials (logon name and password for the application requested).
3. After the user supplies the user credentials, the SSO Client sends the credentials to the CA SSO Server and the SSO Client completes the logon process with the new logon credentials.

Note: Learn mode only functions for users who are authorized to use an application and who have carried out primary authentication. Subsequent logon attempts to the same application by that user automatically uses the credentials previously entered in learn mode.

If the user wants to change their credentials, they can do this by right-clicking on the application button on the SSO Launchbar or selecting the application in SSO Tools and selecting ‘Change Password’. This tells the CA SSO Server to change the credentials, and the next time the application is run the new credentials are returned.

The administrator can clear a user’s credentials by viewing the user’s application list in the Policy Manager (select the User from the data store, click the Application List button, then select the application), and clicking the Update Login Vars button for the application.

Logon Script Maintenance

You should remember that CA SSO logon scripts use and interact with many variables and elements of the computing environment. Changes in the environment affect the operation of logon scripts. For example:

- Changes in hard disk organization that change the location of applications may cause SSO-run commands to fail because the pathname argument will no longer be correct.
- Upgrading an application may result in many changes: new executable name or new logon windows with different titles and field labels. CA SSO extensions that refer to these elements will no longer function as expected.
- Upgrades and changes to operating systems have similar effects.

Therefore, the administrator supporting CA SSO must coordinate personnel responsible for version control and be informed about system environmental changes and application upgrades.

Web-Based Applications

There are three ways to implement CA SSO for web applications:

- Client logon
- Cookie logon – requires a Web Agent
- Browser logon – requires a Web Agent

There are multiple web logon methods because different methods are suited to different web applications and different architectures. You can install all of these methods within the same CA SSO system.

The term web applications also includes restricted web pages.

To define a web application for SSO Client logon, you should define an application in the usual way. The Tcl script you write should launch a browser and navigate to the required web location. You use Tcl to do your logon.

Defining a web application for Cookie logon, requires a CA SSO or CA SiteMinder web agent. For more information on this, contact your Technical Support representative.

Browser logon requires an SSO web agent. For more information on this, contact your Technical Support representative.

Where the Logon Scripts are Stored

The logon scripts are stored as ASCII files and UNICODE on the CA SSO Server host.

Scripts should be saved in the following location:

Windows

\Program Files\CA\Single Sign-On\Server\Scripts

Note: The above locations assume you have installed CA SSO to its default install location.

End User Application Lists

The SSO Client gets the list of SSO-enabled applications from the CA SSO Server and displays it to the user when they log on to CA SSO.

You can configure the CA SSO Server to build all the application lists for all users at non-peak times and store them in a cache. Users can trigger a real-time calculation of their application list directly from the CA SSO Server by clicking Refresh Application List on any of the SSO Client interfaces. You can disable the Refresh Application List function by editing the Client.ini file.

Application Authentication

All application logons supported by CA SSO follow the same overall process. The specific sub-section of application logon that handles the way the user is authenticated to the application is called *application authentication*. CA SSO offers two different methods of application authentication:

- Password authentication. Can be used for applications on any platform (Windows or Mainframe)
- Ticket authentication. Only used for Mainframe applications. Ticket authentication can be broken down into two subsections:
  - PassTickets
  - AppTickets

The application authentication method used for an SSO-supported application is specified in the LOGON_TYPE property of the application’s record. If a value for the LOGON_TYPE property is not specified, the default method used is native SSO password.

The application authentication method used for an SSO-supported application is specified by the value of the LOGIN_TYPE property of the application's record in the CA Access Control data store. Each SSO application record in the data store can have only one application authentication method associated with it. If a value for the LOGIN_TYPE property is not specified, the default method used is native SSO password.

Setting Up Password Authentication

The following steps describe how to set up password authentication for an application:

1. Define the application with a Login Type of Password.
2. Link the application to a password policy using the Policy Manager (if required).
3. Authorize users and user groups to the application using the Policy Manager.
4. Write the logon script using Tcl and place it in the ScriptPath in the Registry on the CA SSO Server host.
5. Enter the script name in the application definition's Script File setting, in the Scripting section.
6. Have a user log into the application. The first time the user logs in, check that Learn Mode is activated.
7. During the second and succeeding logons, the user is not prompted for a password.
8. Change the user’s password to check that the logon script and CA SSO Server process the new password correctly.

Application Icons

You can change the way that application icons appear to users in the SSO Client interfaces.

Change Application Captions

When you add applications to CA SSO you can specify the caption that the user sees underneath the application icon to help users identify applications.

To specify a caption

1. On the Policy Manager, navigate to Resources, Application Resources, Application.
2. Right-click the application, and select Properties.
3. Select the General icon from the side bar of the dialog.
4. Type the caption of the application that you want users to see in the Caption field.

   Note: If no caption is specified, the caption defaults to the application name specified in the Policy Manager.

Change Application Icons

When you add applications to CA SSO you can specify the application icon that the user sees to give them a familiar look and feel.

To specify an icon

1. On the Policy Manager, navigate to Resources, Application Resources, Application.
2. Right-click the application, and select Properties.
3. Select the Attributes icon from the side bar of the dialog.
4. Specify the icon path in the Icon File field. This should be the path to the icon on the client computer. Optionally, it could be a path to a shared resource where all the icon files are located.

   Note: When you specify a path to a UNC using the Policy Manager, you must escape each backslash in the path name. For example, \\ssoserver\icons\icon.ico becomes \\\\ssoserver\\icons\\icon.ico.

Where to Get Application Icons

You can use an icon from one of the following locations.

Program EXE or DLL

You can specify an EXE or DLL file. The SSO Client uses the first icon it finds in the EXE or DLL file.

For example, a Notepad icon retrieved from the Notepad.exe file stored on the SSO Client computer: c:\WINDOWS\SYSTEM32\Notepad.exe

You do not need to specify the full path of the EXE or DLL if the program is found in the Windows PATH. In this case the application name and extension is enough, for example, Notepad.exe.

Custom ICO file

You can specify a custom ICO file. You can store this ICO file on a server or the SSO Client computer. The disadvantage of storing the ICO file on a shared server is that it may cause delays for users while the icons are retrieved, particularly if you have a large number of users or a large number of applications.

If you retrieve the ICO file from the server, you must escape all backslashes.

Here is an example of a custom icon retrieved from the SSO Client computer:

```
c:\icons\icon.ico
```

Here is an example a custom icon retrieved from a shared server computer:

```
\\\\sharedserver\\icons\\icon.ico
```

Generic SSO application icon

CA SSO comes with a generic application icon. If you do not specify an alternative, the user sees the generic icon.

It is possible to override the generic application icon that comes with SSO with one of your own. You configure this in the SSO Client.ini file:

DefaultApplicationIconFile

This is the path to the icon file.

DefaultApplicaitonIconIndexIndex

This is the number of the icon in the .exe or .dll

How the Application Wizard Works

When you use the Application Wizard to generate a CA SSO application script, the Application Wizard:

- Prompts you to step through the process that you want to automate for a specific Windows or browser-based application
- Records the login, application navigation, and other configurable events for the particular browser-based or Windows application
- Generates the SSO application script automatically
- Lets you test and save the script

Using the SSO Application Wizard to Create SSO Scripts

This section describes how to use the Application Wizard to automatically create SSO application Tcl scripts for browser-based and Windows applications.

The Application Wizard

The SSO Application Wizard lets you add your applications to SSO without the need to learn Tcl scripting for basic scripting tasks.

Use the SSO Application Wizard to create SSO application scripts for browser-based applications and Windows applications, such as login scripts and scripts that automate post-login tasks. For example, navigating to a specific screen or dialog in an application, or navigating to a specific web page. You can perform more complex application integration or task automation processes by writing Tcl scripts manually.

Note: The SSO Application Wizard is a standalone Windows application that does not require any other SSO components to generate scripts. However, to test Tcl scripts generated by the Application Wizard, you must install the SSO Client on the local computer.

Application Wizard Generated Scripts

The Application Wizard generates Tcl script code that is ready for use by CA SSO. However, you might need to modify some scripts to provide processing for additional application scenarios. Application Wizard scripts are modular and commented, so that you can easily modify and customize them as required.

For more information on the Tcl scripting language, see the *TCL Scripting Reference Guide*.

How You Create Robust and Effective Scripts

To ensure that the scripts you generate are effective and robust and account for any exceptions to the expected flow of operation, we recommend that you use the following strategy:

1. Define your business requirement.

   For example your business requirement could be to automate the process of logging a user into an application, and automatically display your company's year to date figures.

2. Decide what you want the script to do.

   For example, after you define your business requirement, you decide you want your script to:

   - Log a user into the application.
   - Navigate to a specific page in the application and display your company's year to date figures.
   - End if the user enters invalid login credentials.
   - Disable mouse and keyboard input when the script is running.
   - Display a message indicating the reason for the failure if the script fails.
   - End when a specific window in the application appears.

3. Use a flow chart or diagram to document the required tasks that you want the script to perform. For example:

   a. Start the application.

   b. Enter the user's credentials in the applications login page and click OK.

   c. If a dialog indicating the user has entered invalid credentials appears, end the script.

   d. Navigate to a specific page in the application.

   e. Click the button that displays your company's year to date figures.

4. Document any situations that could change the expected flow of operation and stop the automation process.

   For example, the expected flow of operation could change when:

   - The application prompts a user to change an expired password.
   - A user enters an invalid password, and the application prompts a user to re-enter their password.

5. Note any unique text that the script can use to identify each application window that you want to automate.

6. Test all of your cases to ensure the steps in your task exactly match the application's behavior.

Application Wizard Pages

The following section describes the pages in the Application Wizard.

Select Application Features Page

The Select Application Features page lets you specify the major tasks of the process you want to automate.

This page contains the following fields:

Initialization

Performs the navigation tasks required to display the application's login window or page. Select this option if the default page or window is not the login window.

Login

Specifies in which fields on the login page or window the script enters the user's login credentials, and how the script submits the credentials.

Password Expiration

Detects an expired password error.

Prompts the user for a new password.

Verifies the new password satisfies password policies on the CA SSO Server.

Updates the password value in the application and on the CA SSO Server.

Select this option when you want to automate an expired password task in the login process.

Note: This option is applicable if the application supports password expiry.

Set New Password

Automates the application’s password change facility when a user is provisioned with a new password on the CA SSO Server.

Note: This option is applicable if the application supports manual password change.

After Login

Specifies the actions to perform after application login.

For example, you might want to bypass a Message of the day pop-up window, or view the contents of a particular folder after logging onto your web mail account.

Application Windows Page

The Application windows page lets you specify the tasks in the process you want to automate and specify any additional actions that you want to the script to perform at this stage of the automation process.

This page contains the following fields:

Task list

Displays the tasks in the process you want to automate.

Add

Adds a task to the Task list. Use this button to add a task to your automation process.

Remove

Removes a task from the Task List. Use this button to remove a task from your automation process.

Automate Window

Displays the Automating Windows dialog Use this button to specify the steps in the task you are automating.

Additional SSO Action to perform for this task

Specifies the additional action you want to happen at this stage of the automation process.

The drop-down list contains the following items:

None

No additional actions are to be performed during this task.

Prompt user for a new password

Prompts the user to enter a new password at the beginning of this task. This value will be used for the Type new password user action. This should be performed in the task that performs application password expiry.

Notify Server of login

Notifies the CA SSO Server that a successful login has occurred. This should be performed in the task following the login task.

Notify Server of login failure

Notifies the CA SSO Server that a login attempt has failed. This should be performed in the task that has detected a login error.

Notify server of password change

Notifies the CA SSO Server that a successful password change has occurred. This should be performed in the task following the set new password task.

Notify server of password change failure

Notifies the CA SSO Server that a password change attempt has failed. This should be performed in the task that detects a password change error.

Finish automation after this task

Specifies that the script ends at this point in the automation.

Assign a Click Action to an Exact Point in an Application Window

When you automate a window using the Automating window dialog and the wizard cannot identify a control, you can assign a click action to an exact point in the application window. To assign a click action to an exact point in an application window, for example, click a button on a toolbar, or an item on a menu, you can specify the relative x and y coordinates in the application where the script should perform a click action.

To assign a click action to an exact point in an application window

1. In the table at the bottom of the Automating window page in the Application Wizard, click the row where the control you want to automate appears.

   The wizard highlights the control you selected in red in the application.

2. In the User Action drop-down list, select Click exact point.
3. Click the row you selected again.

   The wizard displays Pick in the Input column.

4. In the cell where Pick is displayed, click and hold.

   A crosshair icon appears.

5. Drag-and-drop the crosshair icon to the exact point in the application window you want to assign the click action to.

   The wizard assigns the click action to the relative x and y points in the application window.

More information:

Automating Window Dialog (see page 325)

Automating Window Dialog

The Automating window dialog lets you:

- Identify the window in your Windows application that you want to automate.
- Specify any information displayed on the window or dialog you are automating that allows the script to uniquely identify the window.
- Assign actions to the controls on the window or dialog you are automating.

This dialog contains the following fields:

Crosshair

Lets you identify the window you want to automate by dragging the crosshair icon onto the title bar of your application window.

Up arrow

Promotes the automation step in the order in which the script performs automation steps.

Down arrow

Demotes the automation step in the order in which the script performs automation steps.

Duplicate

Creates a copy of a control, adds it to the list of controls displayed in the table so that you can assign an additional action to the control.

Title

Defines the title of the window you are automating.

Class

Defines the class of the window you are automating.

Text

Specifies the text the script uses to identify the window you are automating.

Show All

Displays all the controls the wizard finds on the page you are automating.

The wizard displays only common controls by default.

This dialog also contains the following table at the bottom of the screen.

Label column

Displays the labels of the controls found on the page you are automating.

Type column

Displays the type of control.

User Action column

Lets you assign an action to a control on an application window. The items displayed in the drop-down list depend on the type of control selected. This drop-down list contains the following items:

Click

Performs a left-click action on the control.

Double-click

Performs a double-click action on the control.

Right-click

Performs a right-click action on the control.

Select

Selects a check box.

Clear

Clears a check box.

Select menu item

Lets you select a menu item from the window's menu bar.

Select by label

Lets you select a tab by its label.

Select item

Lets you select an item in combo box, list box, or an item in a tree by name.

Select by index

Lets you select a tab, list box, or item in a combo box, by its index value.

Minimize

Minimizes a window.

Maximize

Maximizes a window.

Click exact point

Specifies the relative x and y coordinates in the application where the script should perform the click action.

Expand branch

Lets you expand a branch in a tree without selecting it.

Collapse branch

Lets you collapse a branch in a tree without selecting it.

Type keystrokes

Sends keystrokes to the window. For information on keystroke syntax, see the *Tcl Scripting Reference Guide.*

Use this to navigate a window's menu bar, if the Select Menu Item action fails to find the window's menu bar.

Type password

Types an SSO password into a specified text field.

Type user name

Types an SSO username into a specified text field.

Type new password

Types the value of the pending password into a text field.

Note: This user action requires the pending password to be set. You should only use this when you configure the steps for Password Expiration and Set New Password tasks of your automation.

Type hostname

Types the value of the _HOST login variable into a text field.

Type other text

Types an arbitrary text string into a text field. If selected, allows you to specify the text you want to enter in the Input column.

Input column

Lets you define the input action for the user action you selected for the control.

Automate the User Login Process in a Sample Application

This example procedure automates user login in a sample Building Access Manager application. Use this example to see how to automate the user login process.

To automate the user login process

1. Select Start, Programs, CA, Single Sign-On, CA Single Sign-On Application Wizard.

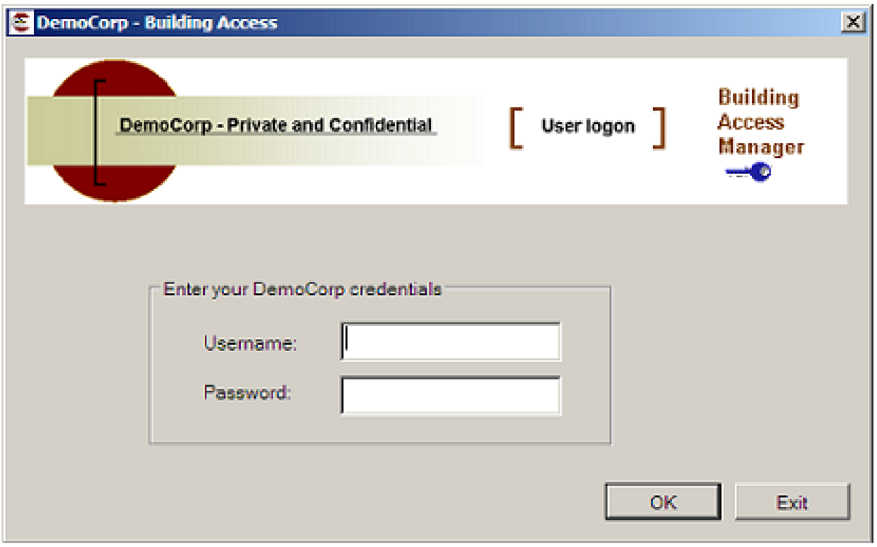
   The Application Wizard starts.

2. Select Windows based application, and then click OK.

   The Select the application to automate page appears.

3. Start the Building Access Manager.

   The User logon window of the Building Access manager appears.



DemoCorp - Building Access

DemoCorp - Private and Confidential

[User logon]

Building Access Manager

Enter your DemoCorp credentials

Username:

Password:

4. From the Select the application to automate page in the wizard, drag-and-drop the crosshair icon onto the title bar of the User Logon window in the Building Access Manager application.

   The wizard populates the Application name and Path to executable fields on the Select the application to automate page.

5. Click Next.

   The Select application features page appears.

6. Select the features of Building Access Manager you want to automate.

7. Define the tasks in the user login process.

8. Automate the user login task.

9. Automate the user login failure task.

   The Select general application options page appears.

10. Complete the fields on the Select general application option page to customize the way your script works.

    The parameters you select affect the execution of the final script.

    Click Next.

    The View or Test automation script page appears.

11. Click Test Script.

    The Tcl Interpreter runs the script.

    The script prompts you to enter an SSO username and password.

12. Click Save Script and select the file where you want to save the script.

    The wizard saves the script in the file you specified.

    Click Finish.

    You have completed automating the user login process.

More information:

Select Application Features Page (see page 322)
Define the Tasks in the User Login Process of a Sample Application (see page 335)
Automate the User Login Task in a Sample Application (see page 330)
Automate the User Login Failure Task in a Sample Application (see page 332)
Select General Application Options Page (see page 344)

Automate the User Login Task in a Sample Application

This example procedure shows you how to automate the user login task in the user login process of a sample Building Access Manager application.

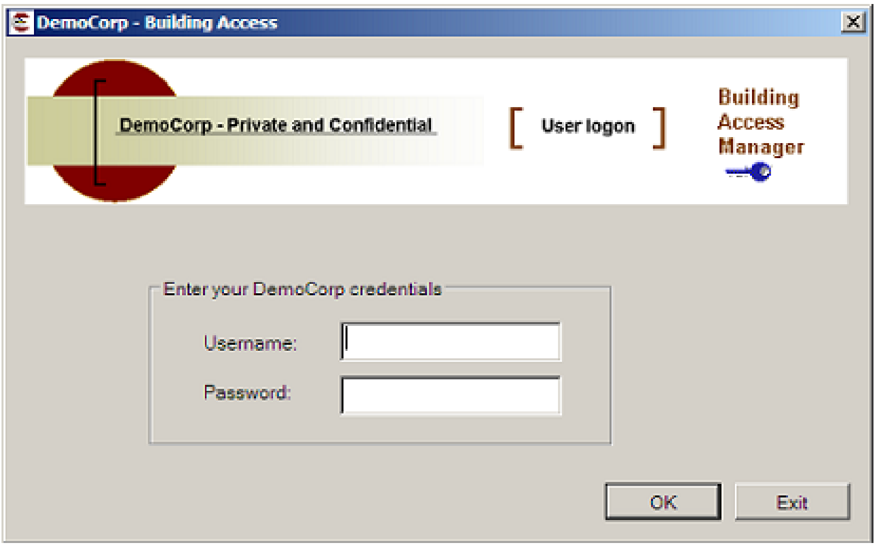
Note: This procedure assumes that you have defined the tasks in the user login process and created a task called User Login in the Application Windows page.

To automate the user login task

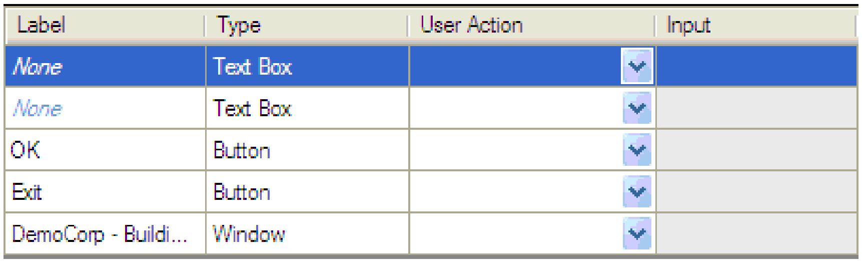
1. In the Task list in the Application windows page, select User Login, and then click Automate window.

   The Automating window: User Login dialog appears.

2. From the Automating window: User Login dialog, drag-and-drop the crosshair icon onto the title bar of the User logon window of the Building Access Manager.


DemoCorp - Building Access
DemoCorp - Private and Confidential
[User logon]
Building Access Manager
Enter your DemoCorp credentials
Username:
Password:
OK
Exit

   The wizard displays the controls found on the User Logon window in a table at the bottom of the Automating window dialog.



Label	Type	User Action	Input
None	Text Box		
None	Text Box		
OK	Button		
Exit	Button		
DemoCorp - Buildi...	Window		

3. On the Automating window dialog, select the Text check box, then type the following text in the Text field:

   Enter your DemoCorp credentials

   The script uses this text to uniquely identify the User logon window of the Building Access manager application. Use this option if your application has multiple windows with the same title.

4. In the User Action column, assign the following actions to the Username, Password and OK controls. Select the following actions from the drop-down list and assign them to the controls respectively:

   - Type username
   - Type password
   - Click

5. Click OK in the wizard.

   The wizard assigns the actions to the controls and the dialog closes.

   You have automated the user login task in your process.

6. Next, you automate the user login failure task in your process.

More information:

Define the Tasks in the User Login Process of a Sample Application (see page 335)
Application Windows Page (see page 323)
Automating Window Dialog (see page 325)
Automate the User Login Failure Task in a Sample Application (see page 332)

Automate the User Login Failure Task in a Sample Application

This example procedure shows you how to automate the user login failure task in the user login process of a sample Building Access Manager application.

Note: This procedure assumes that you have defined the tasks in the user login process and created a task called User Login Failure in the Application windows page.

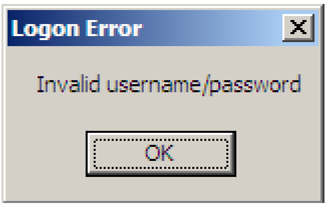
To automate user login failure task

1. In the Task list on the Application windows page, select User Login Failure, and then click Automate Window.

   The Automating window: User Login Failure dialog appears.

2. In the Building Access Manager, enter *invalid* login details in the User Logon window, and then click OK.

   The Logon Error dialog appears.



Logon Error

Invalid username/password

3. From the Automating Window: User Login Failure dialog, drag-and-drop the crosshair icon onto the title bar of Logon Error dialog.

   The wizard displays the title and class of the dialog in the Window Identification section of the Automating window dialog, and displays the controls found on the dialog in the table at the bottom of the page.

4. On the Automating window, select the Text check box, then in the Text field, type the following text:

   Invalid username/password

   The script uses this text to uniquely identify the Logon Error dialog.

5. In the Label column, select the OK control, and then in the User Action drop-down list, select the click action, then click OK.

   The wizard assigns the click action to the OK button.

6. Select the Finish automation after this task check box, and then click.

   This check box specifies that the script ends if a user enters invalid login credentials and the Logon Error dialog appears.

7. Click Next.

   You have automated the failed user login task in your process.

   The Select general application options page appears.

8. Complete the fields on the Select general application options page.

   The parameters you select affect the execution of the final script.

   Click Next.

   The View or Test automation script page appears.

9. Click Test Script.

   The Tcl Interpreter runs the script.

   The script prompts you to enter your SSO user name and password.

10. Click Save Script and select the file where you want to save the script.

    The wizard saves the script in the file you specified.

    Click Finish.

    You have completed automating the user login process.

More information:

Define the Tasks in the User Login Process of a Sample Application (see page 335)
Application Windows Page (see page 323)
Automating Window Dialog (see page 325)
Select General Application Options Page (see page 344)

Application HTML Forms Page

The Application HTML Forms page lets you specify the tasks in the process you want to automate and specify any additional actions that you want to the script to perform at this stage of the automation process.

This page contains the following fields:

Task list

Displays the tasks in the process you want to automate.

Add

Adds a task to the Task list. Use this button to add a task in your automation process.

Remove

Removes a task from Task List. Use this button to remove a task from your automation process.

Automate Form

Displays the Automating form dialog.

Additional SSO Action to perform for this task

Specifies the additional action you want to happen at this stage of the automation process.

The drop-down list contains the following items:

None

No additional actions are to be performed during this task.

Prompt user for a new password

Prompts the user to enter a new password at the beginning of this task. This value will be used for the Type new password user action. This should be performed in the task that performs application password expiry.

Notify Server of login

Notifies the CA SSO Server that a successful login has occurred. This should be performed in the task following the login task.

Notify Server of login failure

Notifies the CA SSO Server that a login attempt has failed. This should be performed in the task that has detected a login error.

Notify server of password change

Notifies the CA SSO Server that a successful password change has occurred. This should be performed in the task following the set new password task.

Notify server of password change failure

Notifies the CA SSO Server that a password change attempt has failed. This should be performed in the task that detects a password change error.

Finish automation after this task

Specifies that the script ends at this point in the automation.

Define the Tasks in the User Login Process of a Sample Application

This example procedure defines tasks to automate in the user login process of a sample Building Access Manager application. Use the following example to see how to define the tasks in the user logon process.

This example assumes that you have:

- Started the Application Wizard
- Started the Building Access Manager
- Identified Building Access Manager as the application you want to automate
- Selected the features of Building Access Manager you want to automate

To define the tasks in the user login process

1. In the Application windows page of the wizard, create a task for each task in the login automation process. Do the following

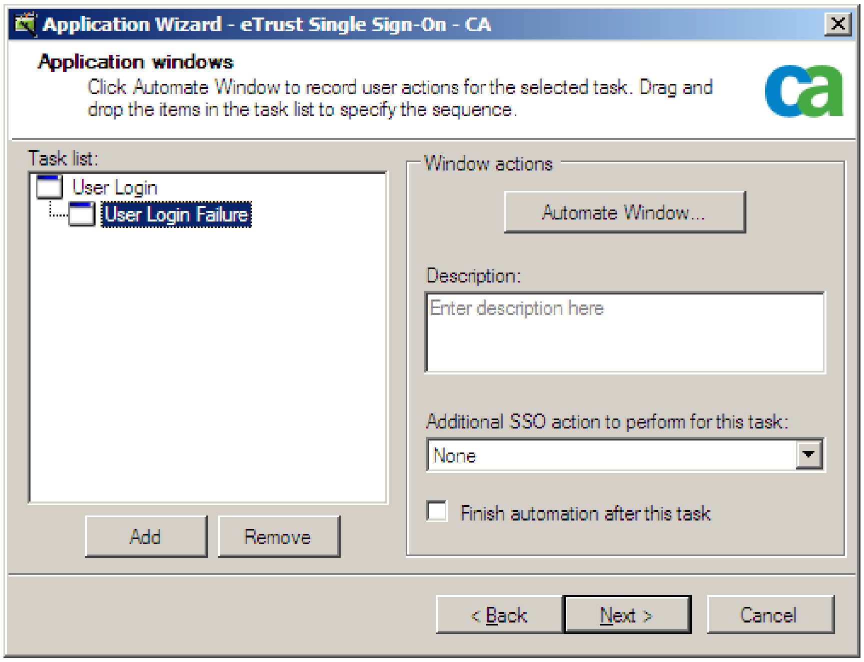
   a. Click Add.

      The Application Wizard adds a task to the Task list on the Application windows page.

   b. Click the task and name it User Login.

   c. Click Add.

   d. Click the task and name it User Login Failure.



Application Wizard - eTrust Single Sign-On - CA

Application windows

Click Automate Window to record user actions for the selected task. Drag and drop the items in the task list to specify the sequence.

Task list:

- User Login
  - User Login Failure

Window actions

Description: Enter description here

Additional SSO action to perform for this task: None

□ Finish automation after this task

The tasks you defined for each task in the user login process appear in the Task list.

2. Next, you automate the user login task in your process.

More information:

Assign a Click Action to an Exact Point in an Application Window (see page 324)
Automate the User Login Process in a Sample Application (see page 328)

Create an SSO Application Script for a Windows Application

To generate SSO application Tcl scripts for Windows applications, you can use the Application Wizard.

To create an SSO application script for a Windows application

1. Start the application you want to automate.
2. Select Start, Programs, CA, Single Sign-On, CA Single Sign-On Application Wizard.

   The Application Wizard starts.
3. Select Windows application and click then Next.

   The Select the application to automate page of the wizard appears.
4. Drag-and-drop the crosshair icon onto the title bar of the window application that you want to automate.

   The wizard populates the Application name and Path to executable fields.
5. (Optional) Define any command line arguments that are required when the application starts.

   Click Next.

   The Select Application features page of the wizard appears.
6. Select the appropriate check boxes to specify the major tasks in the process you want to automate.

   Click Next.

   The Application windows page of the wizard appears.
7. Define any additional major tasks in the process you want to automate, for example, a user login failure. Do the following:

   a. Click Add.

      The Application Wizard adds a task to the Task list on the Application windows page.

   b. Click the task and name it accordingly, for example, User Login Failure.

   c. Repeat the previous two steps for each task in the process you want to automate.
8. Automate each task in the process.
9. (Optional) Select an option from the Additional SSO action to perform for this task drop-down list.

   This drop-down list specifies the additional actions you can specify at this stage of the automation.

10. (Optional) Select the Finish automation after this task check box.

    This check box specifies that the script ends at this point in the automation. For example, you could specify that the script ends if a user enters invalid login credentials.

    Note: For optimal script performance you must select this option for at least one task in the automation process.

11. Click Next.

    The Select general application options page appears.

12. Complete the fields on the Select general application options page to customize the way your script works.

    The parameters you select affect the execution of the final script.

    Click Next.

    The View or Test automation script page appears.

13. Click Test Script.

    The Tcl Interpreter runs the script.

    The script prompts you to enter an SSO username and password.

    Note: In order to test the script through the Tcl Interpreter, the SSO Client must be installed on the local machine.

14. Click Save Script.

    The wizard saves the script in the directory you specified.

15. Click Finish.

More information:

Select Application Features Page (see page 322)
Application Windows Page (see page 323)
Automate Each Task in the Process (see page 339)
Select General Application Options Page (see page 344)

Automate Each Task in the Process

To automate each task in your automation process and assign actions to the controls on a window or dialog, you can use the Automating window dialog in the Application Wizard.

Note: This procedure assumes that you started the Application Wizard and selected the major tasks you want to automate.

To automate each task in the process

1. On the Application windows page of the Application Wizard, select the task you want to automate in the Task list and then click Automate Window.

   The Automating window dialog appears.

2. Navigate to the window or dialog of the application you want to automate.
3. In the Application Wizard, drag-and-drop the crosshair icon onto the title bar of the window or dialog that you want to automate.

   The wizard:

   - Displays the title and class of the window or dialog you want to automate in the title and class fields.
   - Displays the controls found on the window or dialog you selected in a table at the bottom of the Automating window page.

     Note: The wizard does not display non-standard or customized controls by default. To display these, click Show All.

4. (Optional) To distinguish between multiple windows in the application that you want to automate that may have the same title, or are otherwise indistinguishable, do the following:

   a. In the application you are automating, select a text string on the window or dialog that the script can use to uniquely identify the window or dialog. For example, a label identifying a button, or the label used to identify a group of options on the window.

   b. Select the Text check box on the Automating window page.

   c. Type the text you identified in the Text field.

      The script uses this text to identify the window or dialog that you want to automate.

5. In the table on the Automating window page, do the following:

   a. Select the control you want to automate.

      The wizard highlights the corresponding control you selected in red on the window or dialog you are automating.

   b. In the User Action column, select an action to assign to a control from the User Action drop-down list.

   c. Repeat the previous two steps for each control that you want to automate.

6. (Optional) When the wizard cannot identify a control, you can assign a click action to an exact point in an application window.

7. (Optional) To assign an additional action to a control do the following:

   a. Select a control in the table on the bottom of the Automating window page, and then click Duplicate.

      A copy of the control appears in the table on the bottom of the Automating window page.

   b. Select an action to assign to each copy of the control from the User Action drop-down list.

      Note: You cannot delete a control from the table. If you want the script to ignore a control, do not assign an action to the control. The script ignores controls that do not have an action assigned to them.

8. (Optional) To change the order in which the script performs the automation actions, select the control in the table, and then click the up or down arrows.

   The script performs the user actions in the order they appear in the table.

   Click OK.

   The dialog closes.

9. Repeat this procedure for each major task in the process you defined earlier.

More information:

Automating Window Dialog (see page 325)
Assign a Click Action to an Exact Point in an Application Window (see page 324)

Example: A Windows Application User Login Automation

The example detailed in the following topic shows you how you can create a script that automates the login process for a company's Windows application. The name of the hypothetical application used in this example is Building Access Manager.

This example assumes that you followed the recommendations on how you create robust and effective scripts and as a result determined that you want your script to automate the following tasks in the login process:

- User login
- User login failure

Once you create the script for this example using the wizard, it performs the following tasks:

- Start the Building Access Manager
- Enter the users login credentials into the password and username fields on the User logon page of the Building Access Manager
- Click the OK button on the User logon page of the Building Access Manager
- End if invalid login credentials were used

More information:

How You Create Robust and Effective Scripts (see page 321)

Automating Form Dialog

The Automating form dialog lets you:

- Load the pages in your browser-based application that you want to automate and display them in the dialog's embedded browser.
- Specify the information displayed on the web page you are automating that allows the script to uniquely identify the window.
- Assign actions to the controls on the web page you are automating.

This dialog contains the following fields:

Go

Loads the web page you are automating in the embedded browser.

Clear History

Deletes the browsing history.

Stop

Stops loading the web page.

Up arrow

Promotes the automation step in the order in which the script performs automation steps.

Down arrow

Demotes the automation step in the order in which the script performs automation steps.

Duplicate

Creates a copy of a control, adds it to the list of controls displayed in the table and lets you assign an additional action to the control.

Page Title

Displays the title of the window you are automating.

Unique Text

Specifies the text the script uses to identify the window you are automating.

Show Links

Displays all hyperlinks the wizard finds on the page you are automating.

Block pop-ups

Lets you blocks pop-up windows such as scripting errors, security warnings and prompts for authentication that interfere with the automation process.

Note: This check box is only applicable when you are using the wizard. The generated script does not use this check box.

This dialog also contains the following table at the bottom of the screen:

Type column

Displays the type of control.

Description column

Displays the labels of the controls.

User action column

Lets you assign an action to a control on an web page. The items displayed in the drop-down list depend on the type of control selected. This drop-down list contains the following items.

Select

Selects a check box or option.

Clear

Clears a check box or option.

Follow link

Clicks a hyperlink.

Push button

Clicks a button.

Select item

Lets you specify a list item in combo box to select by name.

Type password

Types an SSO password into a specified text field.

Type user name

Types an SSO username into a specified text field.

Type new password

Types the value of the pending password into a text field.

Note: This user action requires the pending password to be set. You should only use this when you configure the steps for Password Expiration and Set New Password tasks of your automation.

Type hostname

Types the value of the _HOST login variable into a text field.

Type other text

Types an arbitrary text string into a text field. If selected, allows you to specify the text you want to enter in the Input column.

Input column

Lets you define the input action for the user action you selected for the control.

Select General Application Options Page

The Select General Application Options page lets you customize the way your script works.

This page contains the following fields:

Failure timeout

Specifies the maximum number of seconds to wait for each SSO action to complete.

Pause afterwards

Specifies the number of seconds to wait after each SSO action completes. Select this option when you need to debug a script when script performance is affected by timing issues.

Display a status message to indicate that automation is taking place

Displays a message box when the script is running indicating details of the automation progress.

Prevent user input by keyboard and mouse button

Disables any keyboard or mouse activity when the script is running.

Display an error message that will indicate why automation has failed

Displays an error message indicating why automation has failed.

Stop execution of the automation script

Stops execution of the script if the Tcl interpreter detects an error.

Before you Install

This section describes what you need to know before you install the SSO Application Wizard. In addition to the information in this section, always make sure you review your implementation plan and take note of any specific requirements.

Pre-Installation Checklist

Use this checklist to make sure you have performed all pre-installation tasks before you install the Application Wizard.

- Download the Application Wizard from the CA SSO page of Support Online or contact your local CA support representative.
- (Optional) Install Microsoft .NET Framework v2.0.

  Note: If Microsoft .NET Framework v2.0 is not installed on the computer, the Application Wizard install package displays the Microsoft .NET Framework Install Wizard, and prompts you to install it.
- Install the SSO client to test scripts generated by the SSO Application Wizard.

Define the Tasks in the User Login Process of a Sample Browser-based Application

This example procedure defines tasks to automate in the user login process in the sample DemoCorp browser-based application. Use the following example to see how to define the tasks in the user login process.

This example assumes that you have:

- Started the Application Wizard
- Selected the features of the DemoCorp browser-based application you want to automate.

To define the tasks in the user login process

1. In the Application HTML forms page, create a task for each step in the login automation process. Do the following:

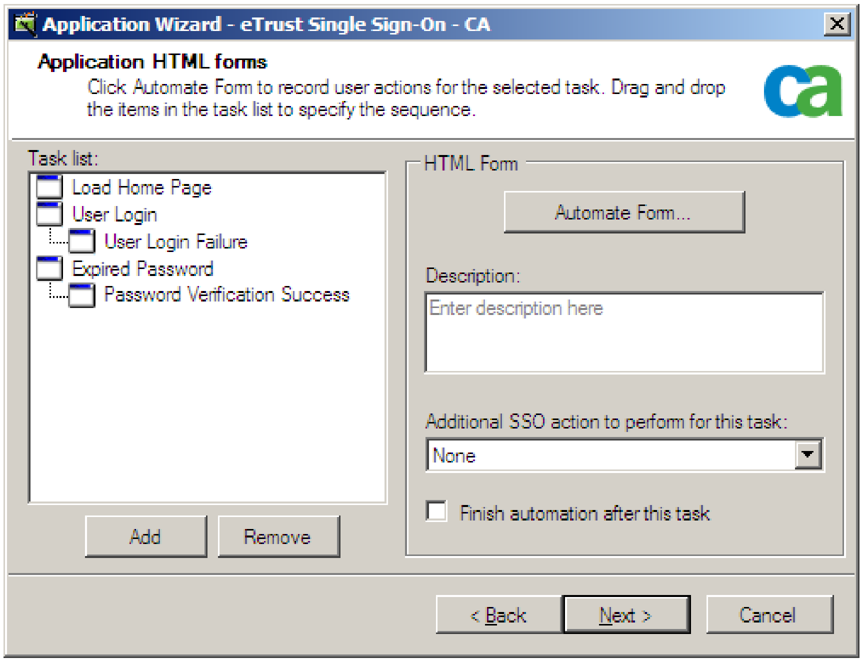
   a. Click Add.

      The Application Wizard adds a task to the Task list on the Application HTML forms page.

   b. Click the task and name it one of the following. Name the tasks in the following order:

      - Load Home Page
      - User Login
      - User Login Failure
      - Expired Password
      - Password Verification Success

   c. Repeat the previous two steps until you have added a task for each step in the automation process.



Application Wizard - eTrust Single Sign-On - CA

Application HTML forms

Click Automate Form to record user actions for the selected task. Drag and drop the items in the task list to specify the sequence.

Task list:

- Load Home Page
- User Login
  - User Login Failure
- Expired Password
  - Password Verification Success

HTML Form

Description: Enter description here

Additional SSO action to perform for this task: None

□ Finish automation after this task

The tasks you defined for each step in the user login process appear in the Task list.

2. Next, you automate the load home page task in your process.

More information:

Application HTML Forms Page (see page 334)
Automate the Load the Home Page Task in a Sample Browser-based Application (see page 349)

Automate the User Login Process in a Sample Application

This example procedure automates user login for a sample browser-based application in the sample DemoCorp browser-based application. Use this example to see how to automate the user login process.

To automate the user login process

1. Select Start, Programs, CA, Single Sign-On, CA Single Sign-On Application Wizard.

   The Application Wizard starts.

2. Select browser-based application, and then click OK.

   The Select application features page appears.

3. Select the features of DemoCorp browser-based application that you want to automate.
4. Define the tasks in the user login process.
5. Automate the load home page task.
6. Automate the user login task
7. Automate the user login failure task.
8. Automate the expired password task.
9. Automate the successful password verification task.
10. Click Next.

    The Select general application options page appears.

11. Complete the fields on the Select general application option page.

    The parameters you select affect the execution of the final script.

    Click Next.

    The View or Test automation script page appears.

12. Click Test Script.

    The Tcl Interpreter runs the script.

    The script prompts you to enter an SSO username and password.

13. Click Save Script and select the file where you want to save the script.

    The wizard saves the script in the file you specified.

    Click Finish.

    You have completed automating the user login process.

More information:

Select Application Features Page (see page 322)
Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Automate the Load the Home Page Task in a Sample Browser-based Application (see page 349)
Automate the User Login Task in a Sample Browser-based Application (see page 351)
Automate the User Login Failure Task in a Sample Browser-based Application (see page 353)
Automate the Expired Password Task in a Sample Browser-based Application (see page 355)
Automate the Successful Password Verification Task in a Sample Browser-based Application (see page 357)
Select General Application Options Page (see page 344)

Automate the Load the Home Page Task in a Sample Browser-based Application

Use the following example to see how to automate the Load DemoCorp Home Page task in the user login process.

Note: This procedure assumes that you have defined the tasks in the user login process and created a task called Load Home Page in the Application HTML forms page.

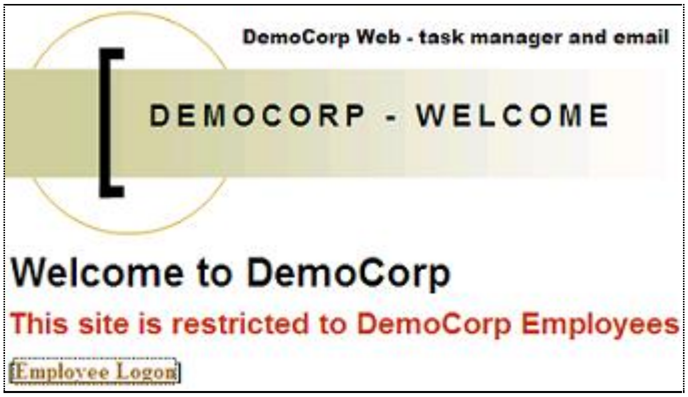
To automate the load home page task

1. In the Task list on the Application HTML forms page, select Load Home Page and then click Automate Form.

   The Automating form: Load Home Page dialog appears.

2. Enter the URL of the DemoCorp home page, then click Go.

   The wizard displays the Demo Corp home page in the embedded browser in the Automating form window.



DemoCorp Web - task manager and email

DEMOCORP - WELCOME

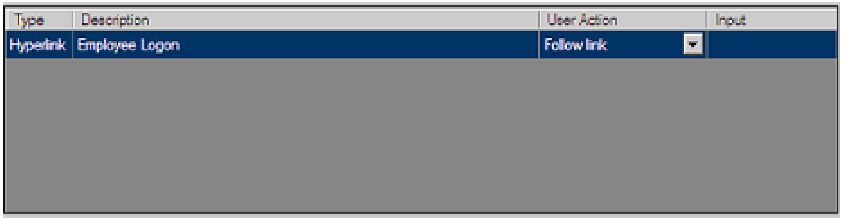
Welcome to DemoCorp

This site is restricted to DemoCorp Employees

The wizard displays the title of the DemoCorp home page in the Page title text box.

3. On the DemoCorp home page, click the Employee Logon hyperlink.

   The wizard displays the Employee Logon hyperlink on the DemoCorp home page in a table at the bottom of the Automating form dialog as shown below and automatically assigns the action Follow link to the control.



Type	Description	User Action	Input
Hyperlink	Employee Logon	Follow link	

4. On the DemoCorp home page, select the following text:

   Employee Logon

5. Click Capture.

   The wizard displays the text in the Unique Text field on the Automating form dialog. The script uses this text to uniquely identify the DemoCorp home page.

6. Click OK in the wizard.

   The wizard assigns the Follow link action to the hyperlink.

   The dialog closes.

   You have automated the load home page task in your process.

7. Next, you automate the user login task of your process.

More information:

Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Application HTML Forms Page (see page 334)
Automating Form Dialog (see page 342)
Automate the User Login Failure Task in a Sample Browser-based Application (see page 353)

Automate the User Login Task in a Sample Browser-based Application

This example procedure shows you how to automate the User Login task in the user login process of the DemoCorp's sample browser-based application.

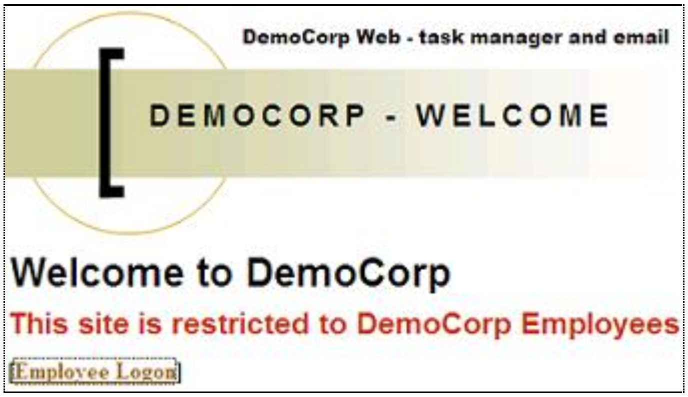
Note: This procedure assumes that you have defined the tasks in the user login process and created a task called User Login in the Application HTML forms page.

To automate the user login task

1. In the Task list on the Application HTML forms page, select User Login and then click Automate Form.

   The Automating form: User Login dialog appears.

   The wizard displays the DemoCorp home page in the embedded browser.



DemoCorp Web - task manager and email

DEMOCORP - WELCOME

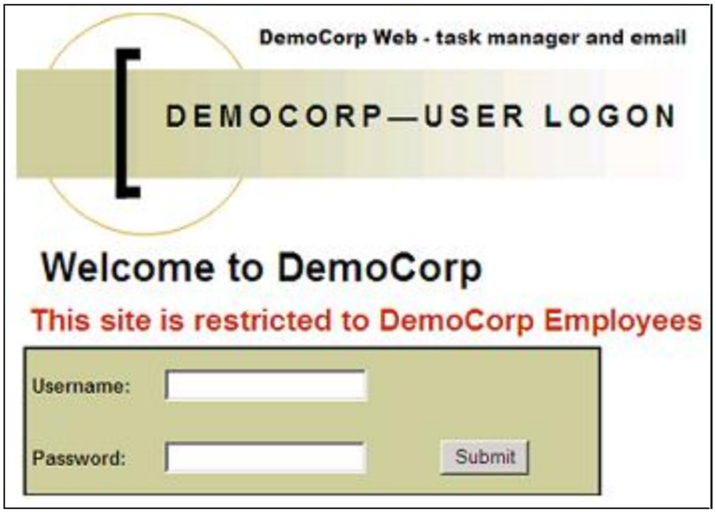
Welcome to DemoCorp

This site is restricted to DemoCorp Employees

Employee Logon

2. Hold down the Ctrl key, and then click the Employee Logon hyperlink.

   The wizard clicks the hyperlink and displays the User Logon page.



DemoCorp Web - task manager and email

DEMOCORP—USER LOGON

Welcome to DemoCorp

This site is restricted to DemoCorp Employees

Username:

Password:

The wizard displays the Username, Password and Submit button controls found on the DemoCorp home page in a table at the bottom of the Automating form dialog and automatically assigns the Type user name, Type password and Push button actions to the controls respectively.

3. On the User Logon web page, select the following text:

   Username

4. Click Capture.

   The wizard displays the text in the Unique Text field on the Automating form dialog. The script uses this text to uniquely identify the page.

5. Click OK.

   The Automating from dialog closes.

   You have automated the User Login task in your process.

6. In the Additional SSO action to perform for this task drop-down list, select Notify server of login.

   When the script runs, the script notifies the CA SSO Server that a successful login has occurred.

7. Next, you automate the user login failure task in your process.

More information:

Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Application HTML Forms Page (see page 334)
Automating Form Dialog (see page 342)
Automate the User Login Failure Task in a Sample Browser-based Application (see page 353)

Automate the User Login Failure Task in a Sample Browser-based Application

This example procedure shows you how to automate the User Login task in the user login process of the DemoCorp's sample browser-based application.

Note: This procedure assumes that you have defined the tasks in the user login process and created a task called User Login Failure in the Application HTML forms page.

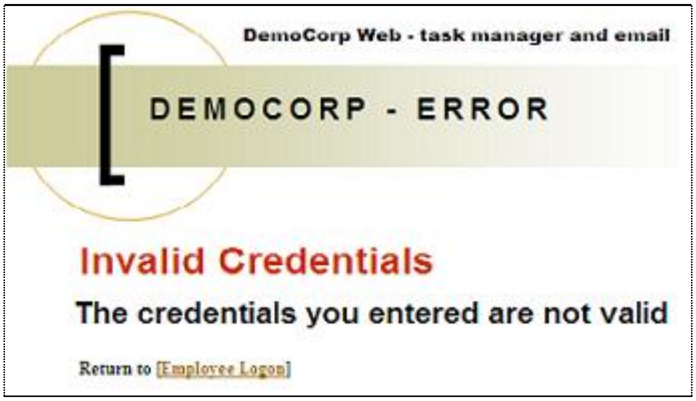
To automate the user login failure task

1. In the Task list on the Application HTML forms page, select User Login Failure and then click Automate Form.

   The Automating form: User Login Failure dialog appears.

2. Navigate to the DemoCorp User Logon page.
3. On the DemoCorp User Logon page, enter *invalid* login details, hold down the Ctrl key, and then click Submit.

   The wizard clicks the Submit button and the Invalid Credentials page appears.



DemoCorp Web - task manager and email

DEMOCORP - ERROR

Invalid Credentials

The credentials you entered are not valid

Return to [Employee Logon]

4. On the Invalid Credentials page, select the following text:

   Invalid Credentials

5. Click Capture.

   The wizard displays the text in the Unique Text field on the Automating form dialog. The script uses this text to uniquely identify the page.

6. Click OK.

   The Automating form dialog closes.

   Note: As you do not want the script to perform any actions on this page, you do not need to assign any actions to the controls on this page.

7. In the Additional SSO action to perform for this task drop-down list, select Notify server of login failure.

   When the script runs, the script notifies the CA SSO Server that a login failure has occurred.

8. Select the Finish automation after this task check box.

   This check box specifies that the script ends if a user enters invalid login credentials and the Invalid Credentials page appears.

   You have automated the User Login failure task in your process.

9. Next, you automate the expired password task in your process.

More information:

Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Application HTML Forms Page (see page 334)
Automating Form Dialog (see page 342)
Automate the Expired Password Task in a Sample Browser-based Application (see page 355)

Automate the Expired Password Task in a Sample Browser-based Application

This example procedure shows you how to automate the Expired Password task in the user login process of the DemoCorp's sample browser-based application.

Note: This procedure assumes that you have defined the tasks in the user login process and created a task called Expired Password in the Application HTML forms page.

To automate the expired password task

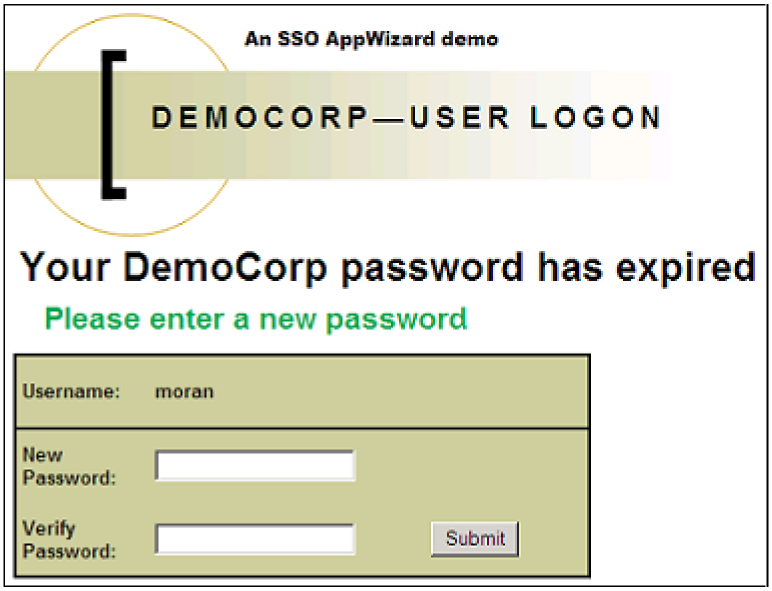
1. In the Task list on the Application HTML forms page, select Expired Password and then click Automate Form.

   The Automating form: Expired Password dialog appears.

2. Navigate to the User Logon Page.
3. On the DemoCorp User Logon page, enter *expired* login credentials, hold down the Ctrl key, and then click Submit.

   The wizard clicks the Submit button and the Expired Password page appears.

   The wizard displays the New Password, Verify Password and Submit button controls found on the Expired Password page in a table at the bottom of the Automating form page. The wizard automatically assigns the Type password and Push button actions to the controls respectively.



An SSO AppWizard demo

DEMOCORP—USER LOGON

Your DemoCorp password has expired

Please enter a new password

Username:	moran
New Password:	
Verify Password:	

4. On the Expired password page, select the following text:

   Your DemoCorp password has expired

5. Click Capture.

   The wizard displays the text in the Unique Text field on the Automating form page. The script uses this text to uniquely identify the page.

6. In the User Action column, assign the Type new password action to the New Password and Verify Password controls.

7. Click OK.

   The Automating form dialog closes.

   You have automated the Expired Password task in your process.

8. Next, you automate the successful password verification task in your process.

More information:

Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Application HTML Forms Page (see page 334)
Automating Form Dialog (see page 342)
Automate the Successful Password Verification Task in a Sample Browser-based Application (see page 357)

Automate the Successful Password Verification Task in a Sample Browser-based Application

This example procedure shows you how to automate the Successful Password Verification task in the user login process of the DemoCorp's sample browser-based application.

Note: This procedure assumes that you have defined the tasks in the user login process and created a task called Successful Password Verification in the Application HTML forms page.

To automate successful password verification task

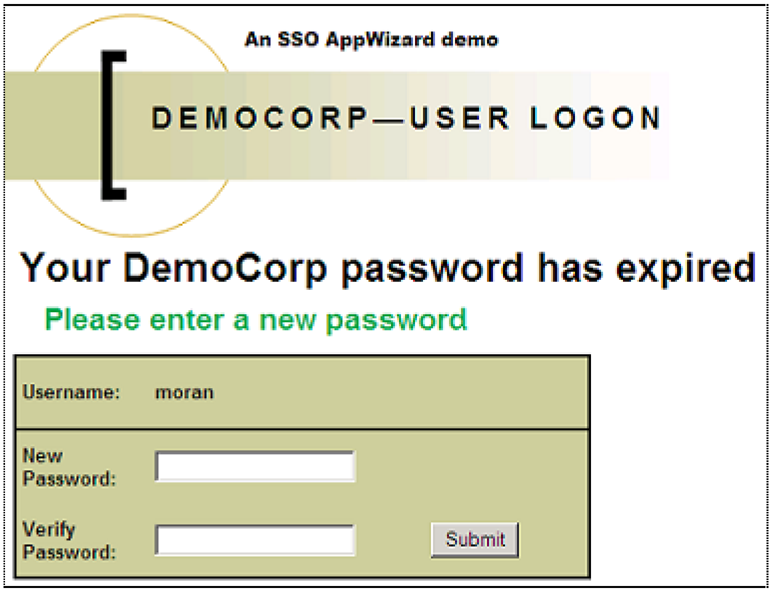
1. In the task list on the Application HTML forms page, select Successful Password Verification and then click Automate Form.

   The Automating form: Successful Password Verification dialog appears.

2. Navigate to the User Logon Page.

3. On the DemoCorp User Logon page, enter *expired* login credentials, hold down the Ctrl key, and then click Submit.

   The wizard clicks the Submit button and the Expired Password page appears in the embedded browser in the Automating form dialog.



An SSO AppWizard demo

DEMOCORP—USER LOGON

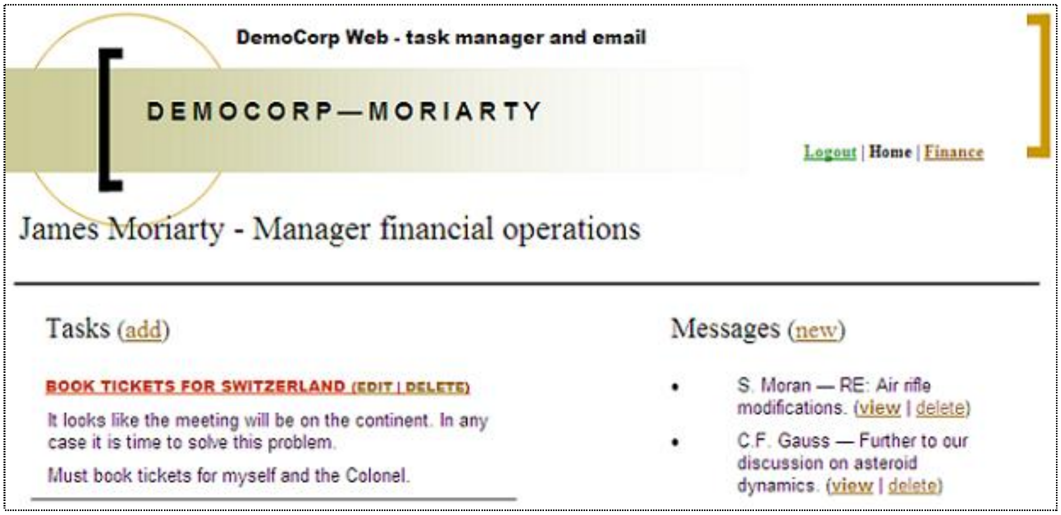
Your DemoCorp password has expired

Please enter a new password

Username: moran

4. In the New Password and Verify Password fields, enter passwords that match, hold down the Ctrl key, and then click Submit.

   The wizard clicks the Submit button and the user's task manager and email home page appears.



James Moriarty - Manager financial operations

Tasks (add)

BOOK TICKETS FOR SWITZERLAND (EDIT | DELETE)

It looks like the meeting will be on the continent. In any case it is time to solve this problem.

Must book tickets for myself and the Colonel.

Messages (new)

- S. Moran — RE: Air rifle modifications. (view | delete)
- C.F. Gauss — Further to our discussion on asteroid dynamics. (view | delete)

5. On the Expired password page, select the following text:

   James Moriarty

6. Click Capture.

   The wizard displays the text in the Unique Text field on the Automating form page. The script uses this text to uniquely identify the web page.

7. Click OK.

   The Automating forms dialog closes.

   Note: As you do not want the script to perform any actions on this page, you do not need to assign any actions to the controls on this page.

8. On the Application HTML forms page, select the Finish automation after this phase check box.

   This check box specifies that the script ends if this page appears.

9. In the Additional SSO action to perform for this task drop-down list, select Notify server of password change.

   When the script runs, the script notifies the CA SSO Server that a successful password change has occurred.

   You have automated the Successful Password Verification task in your process.

10. Click Next.

    The Select general application options page appears.

11. Complete the fields on the Select General Application Option Page.

    The parameters you select affect the execution of the final script.

    Click Next.

    The View or Test automation script page appears.

12. Click Test Script.

    The Tcl Interpreter runs the script.

    The script prompts you to enter your SSO user name and password.

13. Click Save Script and select the file where you want to save the script.

    The wizard saves the script in the file you specified.

    Click Finish.

    You have completed automating the user login process.

More information:

Define the Tasks in the User Login Process of a Sample Browser-based Application (see page 346)
Application HTML Forms Page (see page 334)
Automating Form Dialog (see page 342)
Select General Application Options Page (see page 344)

Install the Application Wizard

To use the Application Wizard you must first install it. To install the Application Wizard, use the Application Wizard install package.

To install the Application Wizard

1. Double-click the setup.exe file in the SSO Application wizard install package.

   The Installation Wizard starts.

2. Follow the wizard prompts, and then click Finish.

   The Installation Wizard installs the Application Wizard onto your computer.

Create an SSO Script for a Browser-based Application

To generate SSO application scripts for browser-based applications, you can use the Application Wizard.

To create an SSO script for a browser-based application

1. Select Start, Programs, CA, Single Sign-On, Single Sign-On Application Wizard.

   The Application Wizard starts.

2. Select Browser-based application and then click Next.

   The Select application features page appears.

3. Select the appropriate check boxes to specify the major tasks of the process you want to automate.

   Click Next.

   The Application HTML forms page appears.

4. Define the additional major tasks in the process you want to automate, for example, user login. Do the following:

   a. Click Add.

      The Application Wizard adds a task to the Application HTML forms page.

   b. Click the task and name it accordingly, for example, User Login.

   c. Add a task for each major step in the process you want to automate.

5. Automate each task in the process.

6. (Optional) Select an option from the Additional SSO action to perform for this task drop-down list.

   This drop-down list specifies the additional actions you can specify at this stage of the automation.

7. (Optional) Select the Finish automation after this task check box.

   This check box specifies that the script ends at this point in the automation. For example, you could specify that the script ends if a user enters invalid login credentials.

   Note: For optimal script performance you must select this option for at least one step in the automation process.

8. Click Next.

   The Select general application options page appears.

9. Complete the fields on the Select general application options page to customize the way your script works.

   The parameters you select affect the execution of the final script.

   Click Next.

   The View or Test automation script page appears.

10. Click Test Script.

    The Tcl Interpreter runs the script.

    The script prompts you to enter an SSO username and password.

    Note: In order to test the script through the Tcl Interpreter, the CA SSO Client must be installed on the local machine.

11. Click Save Script.

    The wizard saves the script in the directory you specified.

12. Click Finish.

More information:

Select Application Features Page (see page 322)
Application HTML Forms Page (see page 334)
Automate Each Task in the Process (see page 363)
Select General Application Options Page (see page 344)

Automate Each Task in the Process

To automate each task in your automation process and assign actions to the controls on a web page, you can use the Application HTML forms page in the Application Wizard.

Note: This procedure assumes that you started the Application Wizard and selected the major tasks you want to automate.

To automate each task in the process

1. On the Application HTML forms dialog, select the task you want to automate in the Task list and then click Automate Form.

   The Automating form dialog appears.

2. On the Automating form dialog, type the URL of the web page you want to automate and then click Go.

   The wizard:

   - Displays the web page in the embedded browser in the Automating form dialog.
   - Displays the controls and features found on the web page in a table at the bottom of the Automating form dialog.

     Note: By default, the wizard does not display hyperlinks in the table on the bottom of the Automating form dialog. To display the hyperlinks the wizard finds on the web page you are automating on the dialog, select the Show Links check box.

   - May assign user actions to controls it detects by default.

     Note: If the wizard detects a form containing an Input field, a Password field and a Submit button, the wizard assigns the Type username, Type password and Push button actions to the controls respectively. This speeds up the automation process and reduces the amount of information that the user needs to specify. We recommend that you check that the actions the wizard assigns to the controls match the desired automation behavior.

3. To navigate to another web page in the application that you want to automate, do the following:

    a. Hold down the Ctrl key, and then click the appropriate navigation control on the web page.

       The wizard performs the default action associated with the control. For example, if you click a hyperlink or button, the wizard follows the hyperlink, or clicks the button.

       If the default action of the control opens the new page in another window, the wizard displays the new page in a new instance of the browser outside the wizard.

    b. To automate this new web page, click Automate Page.

       The wizard displays the new page in the embedded browser in the wizard.

4. (Optional) To distinguish between multiple pages in the browser-based application that you want to automate that may have the same title, or are otherwise indistinguishable, do the following:

    a. On the web page you are automating, select a text string on the web page that the script can use to uniquely identify the page. For example, a label identifying a button.

       Note: The text you select must not span multiple lines. You must select text displayed in the visible part of the page in the embedded browser. The script may not be able to identify text selected outside of this area.

    b. Click Capture.

       The wizard displays the text you selected in the Unique Text field. The script uses this text to identify the web page that you want to automate.

       Note: You may not be able to select text on a button to use as identifying text. Type the text on the button into the Unique Text field.

5. (Optional) Select the Block pop-ups check box.

   The wizard blocks pop-up windows such as scripting errors, security warnings and prompts for authentication that interfere with the automation process.

6. On the web page, do the following:

   a. Select the control you want to automate.

      The wizard highlights the corresponding control you selected in red on the web page and highlights the corresponding control in the table on the bottom of the Automating form dialog. The wizard automatically assigns an appropriate action to the control you selected.

      Note: If you selected a text field, the wizard may not assign an action to the control. Select an action to assign to the control from the User Action drop-down list.

   b. Repeat the previous step for each control that you want to automate.

7. (Optional) To assign an additional action to a control, do the following:

   a. Select a control in the table on the bottom of the Automating form dialog and then click Duplicate.

      A copy of the control appears in the table at the bottom of the Automating form dialog.

   b. Select an action to assign to each copy of the control from the User Action drop-down list.

   Note: You cannot delete a control from the table. If you want the script to ignore a control, do not assign an action to the control. The script ignores controls that do not have an action assigned to them.

8. (Optional) To promote or demote the order in which the script performs automation actions, select the control in the table, then click the up or down arrows.

   The script performs the actions in the order they appear in the table.

   Click OK.

9. Repeat this procedure for each major task in the process you defined earlier.

More information:

Application HTML Forms Page (see page 334)
Select General Application Options Page (see page 344)

Example: A Browser-based Application User Login Automation

The example detailed in the following topic shows you how you can create a script that automates the logon process for a company's browser-based application. The name of the hypothetical company used in this example is DemoCorp.

This example assumes that you followed the recommendations on how you create robust and effective scripts and as a result determined that you want your script to automate the following tasks in the login process:

- Load DemoCorp home page
- User login
- User login failure
- Expired password
- Password verification success

More information:

How You Create Robust and Effective Scripts (see page 321)

What the Script Does

Once you create the script for this example using the wizard, it performs the following tasks:

- Loads the DemoCorp home page
- Clicks the Employee Logon hyperlink on the DemoCorp home page
- Types the user's credentials into the Username and Password fields on the DemoCorp login page
- Clicks the Submit button on the DemoCorp login page
- Display the users task manager and email home page and end
- Ends if a user enters invalid login credentials
- Display the Expired Password page if the user's password has expired

Note: If the Application Wizard is run on Microsoft Vista and a target application requires elevation of privileges to run you must run both the Application Wizard and a target application in elevated mode for the Application Wizard to be able to generate an application script.

Chapter 12: Implementing Session Management

This section contains the following topics:

About Session Management

CA SSO can manage users' SSO sessions. You can:

- Control the number of sessions a user can open concurrently
- Define SSO session behavior, including:
  - What happens if connectivity is lost
  - What happens if the user exceeds their permitted number of sessions
  - An automatic session log out period

Session Management is managed by the CA SSO Server and configured using the SSO Policy Manager. In addition, you can install the SSO Session Administrator which lets you manually view and close user sessions.

When you install CA SSO, the SSO Client is already capable of managing user sessions if you have also installed the SSO GINA or SSO Credential Provider. We recommend that you install and use SSO GINA or SSO Credential Provider, if you want to implement session management.

Note: SSO Credential Provider which is supported on Windows Vista is a replacement for SSO GINA (supported only on Windows XP/2000/2003).

More information:

SSO Credential Provider (see page 23)

Pre-Implementation Considerations

This topic describes what you need to know or do before you implement Session Management.

- Configure the CA SSO Server to enable Session Management.
- Create one or more session profiles and apply them to a user or group using the Policy Manager.
- Install the basic CA SSO components. This includes the following components:
  - SSO Client
  - CA SSO Server
  - Policy Manager
  - Authentication Agent
  - Authentication software installed (if necessary)
- Install the Session Administrator.

  If you are using the SSO Session Administrator, you must create a session management user with administrator rights.
- Synchronize the clocks between the CA SSO Server (or multiple CA SSO Servers if you have a server farm) and the authentication agent machine.

More information:

Install the SSO Session Administrator (see page 370)

Configure the CA SSO Server

This topic describes how to configure the CA SSO Server to enable automatic session management.

To configure the CA SSO Server

1. Launch the Policy Manager.
2. Click Agents and select the default CA SSO Client.
3. Right-click the default CA SSO Client (Agent) and select Properties and then select Session Management.
4. Check the Enable Session Management check box.
5. Complete the remaining fields.
6. Restart the CA SSO Server for the changes to take effect.

Configure Session Management Profiles

Using the Policy Manager, you can set up session profiles that define how the CA SSO Server works with user sessions. Session profiles are groups of settings applied to users or groups of users.

Session profiles include the following settings:

- The number of sessions a user can have open at once
- The result when the user reaches their maximum number of sessions:
  - Terminate the oldest session
  - Terminate the newest session
  - Terminate all sessions
  - Ask the user which of their sessions they want to terminate
  - Reject the registration of the new session – the user is denied log-on
- The result when the system is not used for a time:
  - Define a screen-lock timeout
  - Define a logoff timeout for CA SSO. This timeout must be greater than the screen-lock timeout otherwise it is not considered. CA SSO is logged out after (logoff timeout - screen-lock timeout) time.

Note: When Session Management is turned on in the CA SSO Server, all users have a default policy applied. You can view a user's default policy by going to the Policy Manager, clicking on a user, selecting their session profile list then clicking the **Effective Profile** button.

More information:

Create a Session Profile (see page 383)

Session Termination Settings

There are two ways to configure session management. Also, you can configure a backup method in case the main method fails.

Method 1: Direct Notification (Default)

The CA SSO Server sends a message directly to the SSO Client. This is the faster method of session termination.

Method 2: Terminate Message in Heartbeat Response

The SSO Client sends a regular heartbeat to the CA SSO Server, and the CA SSO Server responds. To terminate a session, the CA SSO Server includes a message in one of its heartbeat responses. This is slower, but it can be used in systems that contain internal firewalls or gateway computers that affect IP addressing, and in particular, prevent direct notification (Method 1).

Backup Method: No Heartbeat Heard

The SSO Client can be configured to take no action, to logout or to terminate a user session if the CA SSO Server does not reply to a certain number of heartbeats. The last two options are useful as a backup in case the main method fails. Also, they prevent connection tampering between the SSO Client and Server.

Install the SSO Session Administrator

This section tells you how to install the SSO Session Administrator. The Session Administrator is the web interface component that allows you to manually view and close user sessions.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the SSO Session Administrator.

- Ensure you meet all system requirements. For information about supported platforms, see the *SSO Readme* file.
- Ensure Java 2 Runtime Environment v1.4.2_09 or later is installed.
- You must have the basic CA SSO components installed and working. This includes the following components:
  - CA SSO Server
  - Policy Manager
  - Authentication Agent
- You must create a session administrator with administration rights.
- Ensure you know which port numbers you want to use for SSL, and the shutdown port. Default information is provided.
- If you plan to install the SSO Session Administrator using silent installation, you need to decide on whether to use a response file or command line options.

More information:

Create a Session Administrator User (see page 387)

Install Using the Wizard

This topic explains how to install the SSO Session Administrator using the Product Explorer. You should use this method to install the SSO Session Administrator on individual computers.

To install the SSO Session Administrator using the wizard

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Configuration Tools, then Session Administrator.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install Using Silent Installation

You can install the SSO Session Administrator silently. This means that you need to provide the information that would normally be supplied by the administrator during installation. This is done using the *setup.exe* command along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the SSO Session Administrator at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Configuration Tools, then Session Administrator.

3. Read the license agreement and note the command line setting for accepting the license agreement.

   You can now install the SSO Session Administrator using silent installation.

4. Open a command prompt and navigate to the SSO Session Administrator folder on the product DVD.

5. From the command prompt, type:

   setup.exe -silent -V LICENSE_VIEWED=*value* {*options*}

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard.

   options

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

setup Command— SSO Session Administrator

The command line parameters for installing the SSO Session Administrator include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-V IS_SELECTED_INSTALLATION_TYPE

Specifies a typical or custom install.

The command has the following format:

```
-V IS_SELECTED_INSTALLATION_TYPE=[Value]
```

Value: Set to typical or custom.

Default: typical.

-V LICENSE_VIEWED

Specify the product license key.

The command has the following format:

```
-V LICENSE_VIEWED=[Value]
```

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

-V SESSADMIN_VIRTUAL_HOSTNAME

Define the Session Administrator host name.

```
-V SESSADMIN_VIRTUAL_HOSTNAME=[Value]
```

Value: The Session Administrator host name.

Default: Localhost

-V SESSADMIN_SSL_PORT

Specify the SSL port number.

```
-V SESSADMIN_SSL_PORT=[Value]
```

Value: The SSL port number.

Default: 8999

-V SESSADMIN_SHUTDOWN_PORT

Specify the shutdown port for the Session Administrator.

```
-V SESSADMIN_SHUTDOWN_PORT=[Value]
```

Value: The shutdown port number.

Default: 8998

Install Using Silent Installation and Response File

Use the following procedures to install the SSO Session Administrator silently using a response file.

Note: You can use a combination of response file and command line options to silently install the SSO Session Administrator. In this case, the command line options will override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the SSO Session Administrator folder on the product DVD.

3. From the command prompt, type:

   ```
   setup.exe -silent {parameters} -options [response file]
   ```

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Defines the name of the response file and location, for example c:\temp\ssorspfile.txt.

More information:

Create a Response File (see page 375)
setup Command— SSO Session Administrator (see page 373)

Create a Response File

Response files record user specific install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the SSO Session Administrator folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record [file name]
   ```

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   A response file is created in the directory specified.

Deploy the CA SSO Client using Unicenter Software Delivery (USD)

You can deploy the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD). To deploy the CA SSO Client and Session Administrator using USD, you need to:

1. Register the software install package with USD.
2. Configure install options.
   - Modify command line options
   - Modify response file options
3. Configure uninstall options.
4. Deploy the software to end users.

Note: The following procedures assume you have USD installed and operational.

1. Register the Software Installer Package with USD

To deploy SSO software using Unicenter Software Delivery (USD), register the software package with USD. Once registered, you need to configure install options prior to deploying to end users.

For more information on registering software with USD, see the *Unicenter Software Delivery Online Help*.

To register the software package with USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Right-click on Software Library and select Register, SD-Package.
3. On the Register SD Package dialog, click Browse and navigate to the appropriate install folder.

   Note: Select the folder that contains the install program, not the file itself.
4. Click Choose and then OK.
5. The software is copied and registered to USD.

2. Modify the Install Package Using USD

Use Unicenter Software Delivery (USD) to modify the software package install options. You can modify:

- Command line options
- Response file options

For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

Modify the Command Line Options

To modify command line options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software package you want to modify.
3. Right-click the install package and click Unseal.
4. Click the sub folders: Procedures, Install.
5. Select the software package and then right-click and select Properties.
6. Click each tab and make the required changes then click OK.

   Note: Ensure all command line information is correct in the Parameters field on the Embedded File tab. For more information on the command line, see the relevant silent install procedure in this chapter.
7. Right-click the software package and select Seal.

   The software package is ready for deployment.

   For more information on modifying install options using USD, see the *Unicenter Software Delivery Online Help*.

Important: You must correctly set the LICENSE_VIEWED parameter to indicate that you agree with the license agreement, otherwise the installation fails.

Modify Response File Options

Use the following procedure to modify the response file options.

To modify response file options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to modify.

3. Right-click the software package and click Unseal.
4. Click the sub folders: Procedures, Install.
5. In the right side panel, right-click the response file and select Properties.
6. Make the required changes then click OK.

   Note: For more information on response file install options, see the relevant setup command options in this chapter.

7. Right-click the software package and select Seal.

   The software package is ready for deployment.

   For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

3. Modify the Uninstall Package Information Using USD

Use Unicenter Software Delivery (USD) to modify the software package uninstall options. You can modify the uninstall command line information.

For more information on modifying the software package using USD, see the *Unicenter Software Delivery Online Help*.

To modify command line options

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software package you want to modify.
3. Right-click the install package and click Unseal.
4. Click the sub folders: Procedures, Uninstall.
5. Select the software package and then right-click and select Properties.
6. Click each tab and make the required changes then click OK.

   Note: Ensure all command line information is correct in the Parameters field on the Embedded File tab.

7. Right-click the software package and select Seal.

   The software package is ready for deployment.

More information:

Uninstall Using Unicenter Software Delivery (USD) (see page 148)

4. Deliver the Software

To deploy SSO software using Unicenter Software Delivery (USD), you must first register the software with Unicenter Software Delivery.

The following procedure guides you through deploying software to a single user using USD. For more information on registering and deploying software using USD, see the *Unicenter Software Delivery Online Help*.

To deploy software to an end user

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and select your software package.
3. Right-click the install package and select Copy.
4. Click All Computers and Users.
5. In the right hand panel, select the computer you want to deploy the software to.
6. Right-click and select Paste>Software/Procedures to Schedule Jobs with Default Settings.

   A job container is created under the side menu heading of Job Containers.

   Note: For more deployment options, see the *Unicenter Software Delivery Online Help*.

Uninstall Using Unicenter Software Delivery (USD)

You can uninstall the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD).

Note: The following procedures assume you have USD installed and operational.

To uninstall using USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to uninstall.
3. Click the sub folder Installations.
4. Right-click All Computers and Users, Schedule Configure Job, Uninstall.

   Note: For more uninstall options, see the *Unicenter Software Delivery Online Help*.

Post-Installation Configuration Options

This section explains some of the configuration options you can implement post-installation.

Create a New Certificate

The Session Administrator comes with a generic, automatically generated certificate. We strongly recommend that you create a new certificate immediately after you install the Session Administrator, and install it in the keystore. You can either do this using Keytool or using a commercial certificate generator. Then, use Keytool to install the certificate into the keystore.

Create a Self-Signed Certificate Using Keytool

To create a self-signed certificate

1. Open a command prompt and navigate to the directory where JRE (Java Runtime Environment) or Java SDK is installed.
2. Navigate to the bin directory.
3. At the prompt, type the following:

   ```
   keytool -genkey -alias tomcat -keyalg RSA -keystore MyNewKeystore.keystore
   ```

   where MyNewKeystore.keystore is the name of the new keystore you are about to generate.
4. When prompted, enter the password for the new keystore.
5. When prompted to enter your first and last name, enter the site name of the Session Administrator application. This will make the certificate match the site name.
6. When prompted, enter information about your organizational unit, organization name, and so on.
7. When the entire DN appears, type either yes or no. You cannot type y or n at this prompt.
8. When prompted to use the same password for Tomcat or create a different password, you can choose either option.
9. The new keystore has now been created.
10. Copy the new keystore file to the conf folder in the Session Administrator install directory. For example, C:\Program Files\CA\Single Sign-On\Session Administrator\conf. directory.

11. Stop the “CA Single Sign-On Session Administrator” Windows Service:
    a. Open the services manager (in Windows, this is in the Administrator Tools section of the Control Panel).
    b. Select the “CA Single Sign-On Session Administrator” Windows Service, then stop the service.
12. Open the server.xml file from conf folder in the Session Administrator install directory. For example, C:\Program Files\CA\Single Sign-On\Session Administrator\conf.
13. Make the following changes to the server.xml file:
    - Change the name of the keystore from *sessionKeystore* to the new keystore file you created.
    - Change the password from *changeit* to the new password you set.
14. Restart the service you stopped before:
    a. Open the services manager.
    b. Select the “CA Single Sign-On Session Administrator” Windows Service, then start the service.
15. Open the Session Administrator, log in, and check that the third item on the certificate dialog is checked.

Create a Certificate Using a Certification Authority

These instructions assume that you are familiar with the certificate management and certification authority concepts.

Before you can get a certificate from a Certification Authority, create a Certificate Signing Request (CSR). The CSR is used by the Certification Authority to create a certificate.

For more information, see: http://tomcat.apache.org/

1. Create a local certificate:

   ```
   keytool -genkey -alias tomcat -keyalg RSA -keystore <your_keystore_filename>
   ```

2. When prompted to enter your last and first name, enter the site name. This ensures that the certificate matches the site name.

3. Create a CSR named certreq.csr:

   ```
   keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr -keystore <your_keystore_filename>
   ```

4. Submit it to the Certification Authority (look at the Certification Authority documentation for more information). In return you get a certificate.

5. Import the certificate into your local keystore:

   a. Download a chain certificate from the Certification Authority you obtained the certificate from.

   b. Import the chain certificate into your keystore:

   ```
   keytool -import -alias root -keystore <your_keystore_filename> \ -trustcacerts -file
   <filename_of_the_chain_certificate>
   ```

6. Import your new certificate:

   ```
   keytool -import -alias tomcat -keystore <your_keystore_filename> \ -trustcacerts -file
   <your_certificate_filename>
   ```

Manually Configure Session Timeout Settings

Once you have installed the Session Administrator, you can configure the default session timeout settings. This is set to 30 minutes during installation and can be updated post installation.

Session timeout lets you set the default session timeout (in minutes) for all administrators logged into the Session Administrator. After the time expires, the user is automatically logged out of the Session Administrator.

To manually configure the Session Management timeout setting

1. Navigate to the conf folder in the Session Administrator install directory. For example, C:\Program Files\CA\Single Sign-On\Session Administrator\conf.

2. Open web.xml and locate the following code:

   ```
   <session-config>

   <session-

   timeout>30</session-timeout></session-config>
   ```

3. Update the session timeout value (30) with the new time.

   Note: If the session timeout limit is exceeded, the administrator user is notified that their current session is no longer valid, and that they need to re-enter their login credentials.

Manually Configure the Session Administrator Inactive Interval

Once you have installed Session Administrator, you can manually configure the default Session Administrator inactive interval. This is set to 5 minutes during installation and can be updated post installation.

The inactive interval lets you set the default inactive time after which a user is logged out of Session Administrator.

To manually configure the Session Management Inactive Interval setting

1. Navigate to the WEB-INF folder in the Session Administrator install directory. For example, C:\Program Files\CA\ Single Sign-On\Session Administrator\webapps\SessionAdministrator\WEB-INF.
2. Open web.xml and locate the following code:

   ```
   <init-param>

   <param-name>session-max-inactive-interval</param-name>

   <param-value>5</param-value>

   <description>Set the maximum inactive interval (ie user doesn't click or do anything to application) in minutes for user's http session</description>

   </init-param>
   ```

3. Update the inactive interval value (5) with the new time.

   Note: If the maximum inactive timeout interval is exceeded, the administrator user is notified that their current session is no longer valid, and that they need to re-enter their login credentials.

Create a Session Profile

You can use the Policy Manager to create session profiles. These profiles can then be assigned to users to determine CA SSO session behavior.

Note: Multiple session profiles can be assigned to a single user. When applying more than one profile to the same user, the most restrictive settings apply.

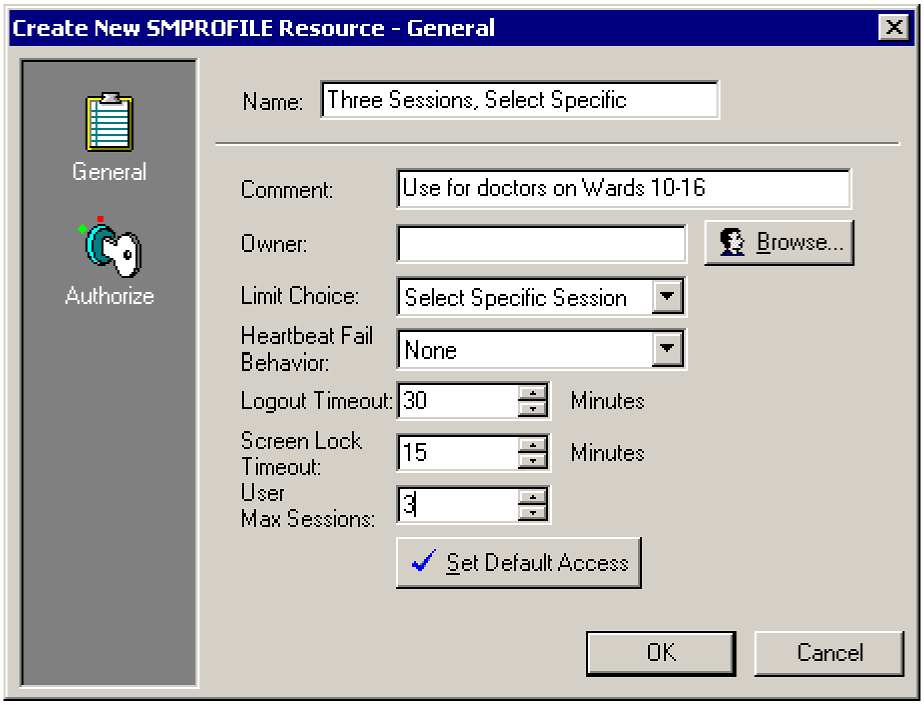
To create a session profile

1. Launch the Policy Manager.
2. Click Resources, Session Resources, Session Profile.

   The list of existing session profiles appears.

3. Right-click anywhere in the list area, and select New.

   The Create New SMPROFILE dialog opens.



Create New SMPROFILE Resource - General

Name: Three Sessions, Select Specific

Comment: Use for doctors on Wards 10-16

Owner:

Limit Choice: Select Specific Session

Heartbeat Fail Behavior: None

Logout Timeout: 30 Minutes

Screen Lock Timeout: 15 Minutes

User Max Sessions: 3

4. In the General dialog, set the behavior for the profile.

   For more information, see SMPROFILE Resource - General Dialog in the *Policy Manager online help*.

5. Click Authorize and set permissions for users or groups to access the new session profile.

6. Click OK.

   The new profile is saved.

Apply a Session Profile to a Single User

You can use the Policy Manager to apply a session profile to a user to control their CA SSO session behavior.

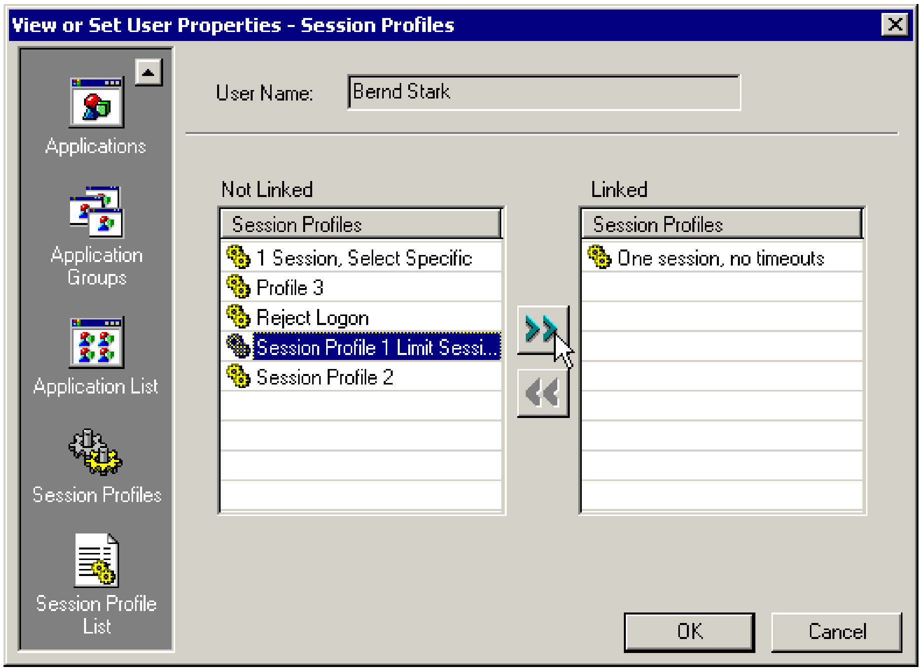
Note: Multiple session profiles can be assigned to a single user. When applying more than one profile to the same user, the most restrictive settings apply.

To apply a session profile to a single user

1. Launch the Policy Manager.
2. Click Users and select the appropriate data store.
3. Double-click a user to open the User Properties dialog.

4. Click Session Profiles.

   A list of all session profiles that can be applied to the user appears.



View or Set User Properties - Session Profiles

User Name: Bernd Stark

Not Linked: Session Profiles
1 Session, Select Specific
Profile 3
Reject Logon
Session Profile 1 Limit Sessi...
Session Profile 2

Linked: Session Profiles
One session, no timeouts

5. Select one or more session profiles and move them to the right so they are linked to the user.
6. Click OK.

   The session profiles are assigned to the user.

Apply a Session Profile to a Group

You can use the Policy Manager to apply a session profile to a group to control their CA SSO session behavior.

Note: Multiple session profiles can be assigned to a group. When applying more than one profile to a group, the most restrictive settings apply.

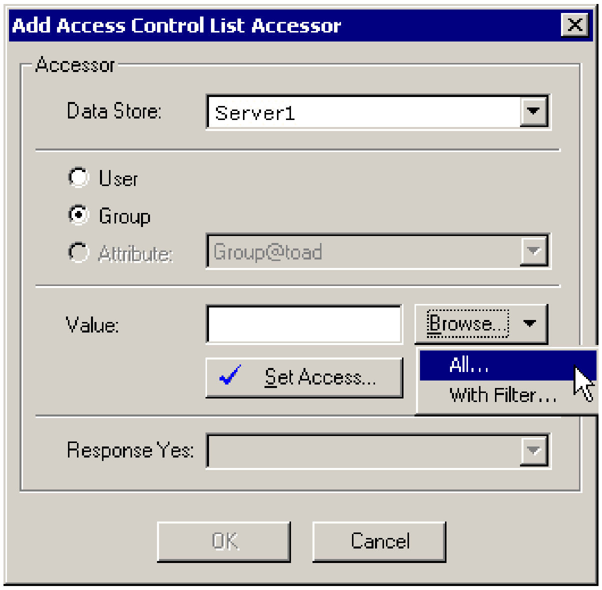
To apply a session profile to a group

1. Launch the Policy Manager.
2. Click Resources, Session Resources, Session Profile, and double-click the session profile name you want to assign a group to.

   The View or Set SMPROFILE Properties dialog appears.

3. Click Authorize.
4. Right-click and select Add.

   The Add Access Control List Accessor dialog appears.



Data Store: Server1

○ User

● Group

○ Attribute: Group@toad

Value:

Response Yes:

5. In the Add Access Control List Accessor dialog, select:
   a. The data store that stores the group.
   b. The Group option.
   c. Group name. If you know the exact name of the group, enter the group name in the Value field.

      Alternatively, click Browse to open the Group Selection dialog, which shows a list of group names. You can either view all group names, or you can filter the list.
6. Click OK when finished.

   The session profile is assigned to the group.

   Note: This procedure can also be used as an alternative for applying a session profile to a user (select the User option at step 5b).

Create a Session Administrator User

Before you can manage user sessions using the SSO Session Administrator, you must set up and assign Session Administrator privileges to a user in the LDAP data store (CA Directory or Active Directory). This lets you launch the Session Administrator and monitor user sessions.

To create a new Session Administrator user

1. In the Policy Manager, create a new user in the LDAP user data store.
2. Run the following selang commands:

   To assign a user administrator rights in ps-ldap, run this command:

   ```
   authorize ROLE ADMIN user_attr("User@ps-ldap") attr_val("cn=<username>") \
   user_dir("ps-ldap") access(Read)
   ```

   The new users are now Session Administrator users meaning they can view and shut down sessions using the SSO Session Administrator.

Update the Locations of the Log Files

There are two kinds of log file for the Session Administrator:

- The Session Administrator’s communications with the CA SSO Server
- The Session Administrator’s inner workings

Also, you can read the logs of the Tomcat server. These logs are written to the C:\Program Files\CA\Single Sign-On\Session Administrator\CATALINA_HOME\logs directory.

Location of the Session Administrator/CA SSO Server Communication Log File

The Session Administrator reports most communication issues (if any) directly into your web browser page, however, if there are some unexpected problems, you can look in the communication log files etWACJavaSDK_C.log and etWACJavaSDK_J.log).

Both files are located in:

```
%SESSION_ADMIN_INSTALLED_DIR%\webapps\SessionAdministrator\log\
```

where

%SESSION_ADMIN_INSTALLED_DIR%

Specifies the installed directory, for example C:\Program Files\CA\Single Sign-On\Session Administrator.

Note: The location of the communication log files is predefined and cannot be changed.

Change the Location of the Session Administrator Log File

Use the following procedure to change the location of the session administrator log file.

To change the log file location

1. Open the following file:

   ```
   %SESSION_ADMIN_INSTALLED_DIR%\log\log4j_config.lcf
   ```

   where

   %SESSION_ADMIN_INSTALLED_DIR%

   Specifies the installed directory, for example C:\Program Files\CA\Single Sign-On\Session Administrator.

2. Find the following line:

   ```
   log4j.appender.session_admin.File=C:/Program Files/CA/Single Sign-On/Session
   Administrator/webapps/SessionAdministrator/log/SessionAdministrator.log
   ```

3. Change the line to refer to a different file location. For example:

   ```
   log4j.appender.session_admin.File=c:\\mylogdir\\mylogfile.txt
   ```

Chapter 13: Communication Modes

This section contains the following topics:

Modes Supported by CA SSO Server and Authentication Agents

CA SSO is enabled to communicate in three different modes. Previous versions of SSO were only capable of supporting one communication mode, now called compatible mode.

Compatible Mode

This mode is the only mode of communication that was used between various components of SSO prior to SSO r12. In this mode, communication between the CA SSO Server and CA SSO Client is achieved using a mixture of ElGamal (for key exchange) and 3DES (for actual message encryption) protocols.

TLS Mode

Federal Information Processing Standards (FIPS) are publicly announced standards developed by the United States Federal government for use by all non-military government agencies and by government contractors. FIPS 140-2 is a standard used to accredit cryptographic modules. SSO r12 implements a new communication mode to meet these standards.

FIPS implementation in SSO involves two major changes:

- Use of AES 256 to encrypt passwords stored in CA SSO Server.
- Use the TLS channel for encryption of data between various SSO components

Note: SSO r12 is capable of communicating in both the compatible and FIPS-only modes simultaneously.

Mixed Mode

Mixed mode provides support for mixed set of clients (for example, clients supporting FIPS communication and legacy clients that do not support FIPS communication) to communicate with SSO r12 Server.

Modes Supported by CA SSO Client and PSA

CA SSO clients and PSAs are enabled to communicate in three different modes. Previous versions of SSO were only capable of supporting one communication mode, now called compatible mode.

Compatible Mode

This mode is the only mode of communication that was used between various components of SSO prior to SSO r12. In this mode, communication between the CA SSO Server and CA SSO Client is achieved using a mixture of ElGamal (for key exchange) and 3DES (for actual message encryption) protocols.

FIPS-only Mode

Federal Information Processing Standards (FIPS) are publicly announced standards developed by the United States Federal government for use by all non-military government agencies and by government contractors. FIPS 140-2 is a standard used to accredit cryptographic modules. SSO r12 implements a new communication mode to meet these standards.

FIPS implementation in SSO involves two major changes:

- Use of AES 256 to encrypt passwords stored in CA SSO Server.
- Use the TLS channel for encryption of data between various SSO components

Note: SSO r12 is capable of communicating in both the compatible and FIPS-only modes simultaneously.

TLS Mode

TLS mode provides support for CA SSO clients to communicate with CA SSO Server and authentication agents in both FIPS-only and compatible modes.

Configuring the CA SSO Client in Different Modes

In the CA SSO Client a DWORD registry value called "CommMode" is provided to configure the communication mode. The key is available under the following branch on the corresponding machines:

HKLM\Software\ComputerAssociates\SingleSignOn\Client

For communication in FIPS-only or dual mode (supports FIPS compatible modes of communication) you must provide the certificate file for the communication. The IdentityFile entry in the auth.ini and client.ini files points to the location of the certificate files:

The values in the client.ini file are related to the communication with the CA SSO Server and the values in auth.ini file are related to the respective authentication agents. The default certificate files are placed in the cfg folder.

Note: If you use custom certificates all instances of these attributes in the auth.ini file and client.ini file have to replaced with the correct path of the certificates.

Configure the CA SSO Client in Compatible Mode

For this mode, the certificate file is used for encryption, mentioned during installation, are required.

To configure the CA SSO Client in compatible mode

1. Set the "CommMode" registry entry to 0 which indicates compatible mode.
2. Exit all the CA SSO client agents such as SSO Tools, status icon, and launch bar.
3. Restart the Client service on CA SSO Client machine.

Configure the CA SSO Client in FIPS-only Mode

For this mode, the certificate file is used for encryption, mentioned during installation is required.

To configure the CA SSO Client in FIPS-only mode

1. Set the "CommMode" registry entry to 1 which indicates FIPS only mode.
2. Set the path of Identity File in the client.ini and auth.ini files.
3. Exit all the CA SSO client agents such as SSO Tools, status icon, and launch bar.
4. Restart the Client service on CA SSO Client machine.

Configure the CA SSO Client in Dual Mode

Dual mode supports FIPS compatible modes of communication. For this mode, the required certificate (in .pem format) must be available in the value %InstallDir%\cfg directory. In the case of Custom certificates, the absolute paths to the Certificate file must be provided as value for IdentityFile in the auth.ini and client.ini files

To configure the CA SSO Client in Dual mode

1. Set the "CommMode registry" entry to 2 which indicates TLS mode.
2. Set the paths of Identity File in the client.ini and auth.ini files.
3. Exit all the CA SSO client agents such as SSO Tools, status icon, and launch bar.
4. Restart the CA SSO Client service on CA SSO Client machine.

   Note: If this configuration is done after installation, you must edit the following entries in the corresponding files %InstallDir%/cfg/Client.ini and Auth.ini.

   ```
   IdentityFile=<FULL_PATH_TO_THE_IDENTITY_FILE_IN_PEM_FORMAT>
   ```

For example:

```
IdentityFile=c:\Program files\CA\Single Sign-on\cfg\sso_rootcert.pem
```

Configuring the CA SSO Server in Different Modes

A new Configuration Property called "CommMode" has been added to the server. "CommMode" contains the configured communication mode value (0,1,2) and the property is located in Access Control. Issue the following command at the selang prompt to show the current value of "CommMode":

```
showRes PSCONFIGPROPERTY CommMode@ssod
```

or

```
sr PSCONFIGPROPERTY CommMode@ssod
```

To change the mode, issue the following command at the selang prompt:

```
er PSCONFIGPROPERTY CommMode@ssod gen_prop(value) gen_val(0 or 1 or   2)
```

Alternately, this can be changed from the Policy Manager by navigating to PolicyServerSettings > Communication and editing the CommMode setting.

The CA SSO Server listens on default port 13980. In FIPS mode, the default SSL port is 13981. In dual mode, the CA SSO Server supports both FIPS and non-FIPS modes of communication. In dual mode both ports are used. The ports configured can be seen using the following two commands:

```
showRes PSCONFIGPROPERTY PortNumber@ssod
showRes PSCONFIGPROPERTY SSLPortNumber@ssod
```

To change the mode, issue the following command at the selang prompt

```
er PSCONFIGPROPERTY PortNumber @ssod gen_prop(value) gen_val(<portnumber>)
er PSCONFIGPROPERTY SSLPortNumber @ssod gen_prop(value) gen_val(<sslportnumber>)
```

Alternately, this can be changed from the Policy Manager by navigating to PolicyServerSettings > Communication and editing the PortNumber and SSLPortnumber settings.

Configure the CA SSO Server in Compatible Mode

To configure the CA SSO Server to run in Compatible mode

1. Set the CommMode property to 0 which indicates Compatible mode. This can be done using one of the following methods:

   - Using selang

     Open Selang command prompt and execute the following command:

     ```
     editres PSCONFIGPROPERTY CommMode@ssod gen_prop(value) gen_val(0)
     ```

     or alternately,

     ```
     er PSCONFIGPROPERTY COmmMode@ssod gen_prop(value) gen_val(0)
     ```

   - Using Policy Manager

     In Policy Manager, go to Communication Settings in Policy Server settings and change the CommMode property to compatible mode.

2. Change the communication mode of CA Access Control to non-FIPS by editing the following entries in the registry key HKLM\Software\ComputerAssociates\AccessControl\Crypto:

   Communication mode

   Set the value of this key to non_ssl

   FIPS_only

   Set the value to 0.

3. Restart the CA Access Control services and then restart the CA SSO Server service.

   The CA SSO Server is configured in compatible mode.

Configure the CA SSO Server in TLS Mode

To configure the CA SSO Server to run in TLS mode

1. Set the "CommMode" property to 1 which indicates FIPS only mode. This can be done using one of the following methods:

   - Using selang

     Open Selang command prompt and issue the following command:

     ```
     editres PSCONFIGPROPERTY CommMode@ssod gen_prop(value) gen_val(1)
     ```

   - Using Policy Manager

     In Policy Manager, go to Communication Settings in Policy Server settings and change the "CommMode" property to TLS.

2. Change the communication mode of CA Access Control to FIPS by editing the following entries in the registry key HKLM\Software\ComputerAssociates\AccessControl\Crypto:

   Communication mode

   Set the value of this key to fips_only

   FIPS_only

   Set the value to 1.

3. Restart CA Access Control services and then restart the CA SSO Server service.

Configure the CA SSO Server in Mixed Mode

To configure the CA SSO Server to run in mixed mode

1. Set the "CommMode" property to 2 which indicates compatible and TLS communication mode. This can be done using one of the following two methods:

   - Using selang

     Open selang command prompt and execute the following command:

     ```
     editres PSCONFIGPROPERTY CommMode@ssod gen_prop(value) gen_val(2)
     ```

   - Using Policy Manager

     In Policy Manager, go to Communication Settings in Policy Server settings and change the CommMode field to mixed.

2. Restart the CA SSO Server service.

Configure the CA SSO Server in TLS Mode after Upgrading from r8.1

After upgrading the CA SSO Server and CA SSO Client to r12, the default mode of operation is "Compatible Mode". This is the legacy (r8.1) method of communication.

After upgrading the CA SSO Server, to change the mode of operation to FIPS-only mode:

1. Copy the default certificates of CA Access Control from the DVD to

   C:\Program Files\CA\Access Control\data\crypto directory

   The two files are sub.key and sub.pem. They reside in the sample_certs directory in the product DVD.

2. Using selang, set the mode of operation of the CA SSO Server and reset the password of all admin accounts (for example, PS-ADMIN)

   ```
   selang>er PSCONFIGPROPERTY CommMode@ssod gen_prop(value) gen_val(1)
   ```

   This command sets the Server mode of operation to FIPS. If the gen_val is set to 2,a mixed mode of operation is used. (The CA SSO Server opens 2 ports, one for the old method of communication and the other for FIPS mode of operation. This is done for backwards compatibility with r8.1 Clients.)

3. Change the password of the admin accounts using the following command at the selang prompt:

   ```
   eu ps-admin password(<new-password>) native-;
   ```

4. Change the registry keys of access control viz.

   HKLM\Software\ComputerAssociates\AccessControl\Crypto -> String Values: Communication_mode and fips_only to be set to fips_only and 1 respectively

5. Restart the machine.

   Note: If you do not want to restart the machine, follow these steps:

   - Restart Access Control secons -s for stopping and seosd -start for starting AC services
   - Restart CA SSO Server from services.msc or net start ssod

After upgrading the CA SSO Client, to change the mode of operation to FIPS only mode:

- Change the registry key viz.

  HKLM\software\ComputerAssociates\SingleSignOn\Client

  CommMode to 1 for FIPS-only Mode

  CommMode to 2 for TLS Mode

Make sure that the path of the key file and the pem file are correct in the registry as well as the client.ini file. The following are the exact keys and values

- In Client.ini, under the section NetworkCommunications, the keys are:
  - IdentityFile = <to be set to the path of the certificate file (.pem file) default cert path is <Installdir>\cfg\sso_rootcert.pem
  - Exit SSOStatus and all other Client Agents (launchbar etc)

- Restart Engine Service (ca CA SSO Client service)

For authentication agents, you do not need to follow any manual steps because during upgrade, the mode selection screen is displayed. However, once installed if you want to change the mode of operation, there are manual steps that must be followed exactly the same way as in CA SSO Client, viz, registry entry for COMMMODE, and Ini file entries. You must then restart the authagent service or restart the machine.

To change the mode of operation of the Policy Manager:

- For SSO Communication:

  HKLM\Software\ComputerAssociates\AccessControl\Client\ClientType-> AZN_COMMMODE value

  - 0 for compatible mode
  - 1 for FIPS mode of operation

- For access control Communication

  HKLM\Software\ComputerAssociates\AccessControl\Crypto->Communication_Mode

  - fips_only for FIPS mode of operation
  - all_modes for compatible mode of operation

  HKLM\Software\ComputerAssociates\AccessControl\Crypto->fips_only value

  - 0 for Compatible mode
  - 1 for FIPS mode

Configure the Policy Manager in Different Modes

The Policy Manager contains Communication modes information in registry keys. Set the following keys as indicated below:

- CA SSO Server Communication

  HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Client\ClientType

  Key: AZN_CommMode

  Value: [0 | 1] For Compatible mode and TLS mode respectively

- Access Control communication

  - Copy the default certificates of CA Access Control from the DVD to the path pointed to by HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\crypto private_key and subject_certificate values.
  - The two files (sub.key and sub.pem) reside in the sample_certs directory on the product DVD.

  HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Crypto

  Key: communication_mode (set to fips_only or non_ssl)

  Value: [fips_only | non_ssl for Compatible mode and FIPS only mode respectively

  Key: fips_only

  Value: [0 | 1] for Compatible mode and TLS mode respectively

Note: Mode 2 is not supported in the Policy Manager as CA Access Control has only 2 modes (FIPS or FIPS and non-FIPS).

Chapter 14: Implementing Citrix Application Migration

This chapter explains how to set up Citrix MetaFrame application migration. Citrix MetaFrame application migration within CA SSO refers to the functionality that lets users transfer an application session launched through CA SSO from one workstation to another. Throughout this chapter we will refer to this functionality just as application migration.

This functionality is only available when you deploy CA SSO within a Citrix MetaFrame client-server environment. Citrix products are sold independently of CA SSO.

This section contains the following topics:

Client Experience of Application Migration

Using application migration, a user can log onto CA SSO on workstation A, open an application from their CA SSO application list, and start working on that application (this is standard CA SSO functionality). The user can then move to workstation B, log onto CA SSO, launch the *same* application, and continue working where they left off because their original session has been transferred (migrated) to the second workstation.

Case study

> A doctor logs into CA SSO on workstation A, and opens the Patient History application from the CA SSO application list. The doctor then gets called away to another ward but wants to continue working on the same patient history in the new ward. The doctor can simply log onto workstation B, launch the same application from his CA SSO application list. The application automatically opens exactly where the doctor was last working.

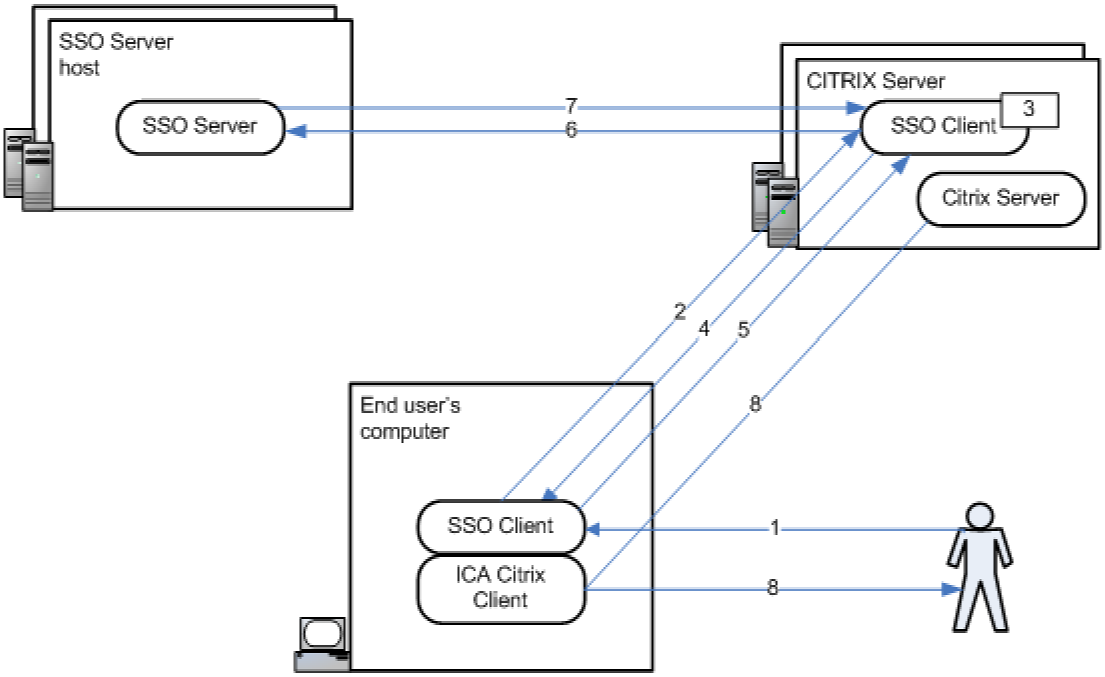
How SSO-Enabled Citrix Applications are Launched

This section describes how an SSO-enabled Citrix application is launched within CA SSO. A Citrix application is an application that runs on the Citrix MetaFrame Presentation server, but is used by someone on the ICA client computer. An *SSO-enabled* Citrix application means that an CA SSO Client has been installed on both the ICA Client computer, to give users the single sign-on experience, but also on the Citrix MetaFrame Presentation server computer where it runs the script to launch the application and supply the logon variables.

This process assumes that the user has already authenticated and has a valid SSO token on the ICA Client computer.

1. The user launches the SSO application from the SSO application list.
2. The client-side CA SSO Client activates an SSO Script that connects to Citrix MetaFrame Presentation Server and tries to run the Citrix Published Application.
3. The Citrix Published Application activates an SSO Script on the Citrix MetaFrame Presentation Server to launch the SSO application.
4. The server-side CA SSO Client requests a valid SSO token from the client-side CA SSO Client.
5. The client-side CA SSO Client sends a valid SSO token to the server-side CA SSO Client.
6. The server-side CA SSO Client then sends the valid SSO token to the CA SSO Server.
7. The CA SSO Server sends the user's logon variables to the server-side CA SSO Client.

8. The server-side CA SSO Client launches the SSO application which the user then sees on the ICA Client computer.



How Application Migration Installation Works

This section is a summary of the steps that you need to set up application migration using CA SSO. The rest of this chapter explains each step in detail. We recommend that you work through this chapter in the order it is written until you understand the process fully:

1. Check that you have all the pre-requisite software, access and logons and fill in the Pre-installation Checklist.
2. Install the CA SSO Client on the Citrix MetaFrame Presentation Server.
3. Install the CA SSO Client on the ICA Client workstation.
4. Write Script A: This is the script you must write to connect to the Citrix MetaFrame Presentation server and run the Citrix Published Application on the ICA Client computer.
5. Write Script B: This is the script you must write to launch the SSO-enabled application on the MetaFrame Presentation Server computer.
6. Define Script A on the CA SSO Server.

7. Define Script B on the CA SSO Server.
8. Create an SSO-enabled published application to run on the MetaFrame Presentation Server computer (this uses Script B).
9. Create another SSO-enabled application to run on the ICA Client computer to connect to the published application on the MetaFrame Presentation Server (this uses Script A).
10. Define the logon credentials for the user for both scripts.

Pre-implementation Considerations

This section outlines all the software, connections and access rights you need to set before you start implementing application migration.

Prerequisite Software

You must have the following software installed and operational before you can set up application migration:

- Citrix MetaFrame Presentation server installed on at least one server machine
- ICA Client installed on at least two workstations
- CA SSO Server installed on a server machine
- Policy Manager installed on a workstation (or server) and connected to the CA SSO Server
- Authentication method (for example, LDAP authentication)

For information about supported platforms, see the *CA SSO Readme.*

Prerequisite Access and Logons

You must have access and logon information set up as follows.

- Administrator logon details for the CA SSO Server
- Administrator logon details for the Citrix Server
- SSO user logon details to SSO
- SSO user logon details for the Citrix MetaFrame Presentation Server

Note: Every SSO user must have unique logon details on the Citrix MetaFrame Presentation Server.

Install Application Migration

This section steps you the process of implementing Citrix Application Migration with CA SSO.

Pre-Installation Checklist

This is a checklist for all the information that you need to implement application migration. Throughout this chapter you will be prompted to write information on this page, so you may want to print it out and write on it.

Be careful to protect password security, for this reason you may choose not to write passwords on this piece of paper.

- CA SSO Server computer name
- CA SSO Server administrator username
- CA SSO Server administrator password
- SSO test user data store name
- SSO test user username
- SSO test user password
- Citrix MetaFrame Presentation server computer name
- Citrix MetaFrame Presentation server administrator username
- Citrix MetaFrame Presentation server administrator password
- Citrix MetaFrame Presentation client test user username
- Citrix MetaFrame Presentation client test user password

The following refers to logon Scripts A and B. You must write a Script A and a Script B for every application that you want users to migrate. We have provided you with example scripts that are listed here and are explained in this chapter.

To help you understand this process, this scenario that follows uses MS Calculator as an example to show every step in the process.   At the end of this chapter you should be able migrate this application. You probably already have MS Calculator installed on your computer as part of a standard MS Windows setup.

Example application name: Calculator
Example Script A name: calc_script_a.tcl
Example Script B name: calc_script_b.tcl

Install the CA SSO Client on an ICA Client Computer

If you want to implement Citrix application migration in conjunction with CA SSO, you must install the CA SSO Client on two different computers. You must install a Citrix ICA client-specific version of the CA SSO Client on the client computer, and a Citrix MetaFrame Presentation Server-specific version of the CA SSO Client on the server computer.

To install the CA SSO Client on an ICA Client computer

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Single Sign-On Client r8.1.

3. Click Install and follow the prompts, but make sure you:

   - Choose Custom installation and choose the options that are appropriate for your environment
   - Choose to install the CA SSO Client on the Citrix ICA Client computer, when prompted
   - Choose authentication method that users must use
   - Create a server set when prompted

   The CA SSO Client is now installed on the ICA Client workstation.

   Note: You can also install the CA SSO Client silently on the ICA Client computer.

More information:

Implementing the CA SSO Client (see page 125)

Install the CA SSO Client on the Citrix MetaFrame Presentation Server

If you want to implement Citrix application migration in conjunction with CA SSO, you must install a Citrix ICA Client-specific version of the CA SSO Client on the client computer, and a Citrix MetaFrame Presentation Server-specific version of the CA SSO Client on the server computer.

Note: If you have several Citrix MetaFrame Presentation Servers in place, be sure to install the CA SSO Client on the one that has the SSO-enabled application installed locally.

To install the CA SSO Client on a Citrix MetaFrame Presentation server computer

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Single Sign-On Client r8.1.

3. Click Install and follow the prompts, but make sure you:

   ■ Choose Custom installation

   ■ Choose Install the CA SSO Client on the Citrix MetaFrame Presentation Server.

   The CA SSO Client will now be installed on the Citrix MetaFrame Presentation Server computer.

   Note: You can also install the CA SSO Client silently on the Citrix MetaFrame Presentation server computer.

More information:

About the CA SSO Client (see page 125)

Create an SSO-Enabled Published Application

This section describes how to configure an application hosted on the Citrix MetaFrame Presentation server so that it can be accessed from a user's CA SSO list on the ICA Client computer.

To create an SSO-enabled published application on the Citrix Metaframe Presentation server

1. Open the Citrix Management Console on the Citrix MetaFrame Presentation Server machine.
2. Choose the Publish Application icon.

   This launches the Application Publishing Wizard.

   Enter the display name and descriptions for the application and press Next. For example:

   Display Name: Calculator

   > This is the name referred to in Script A. This name is not visible to end users.

   Application Description: Calculator

   > The application description is not visible to end users.

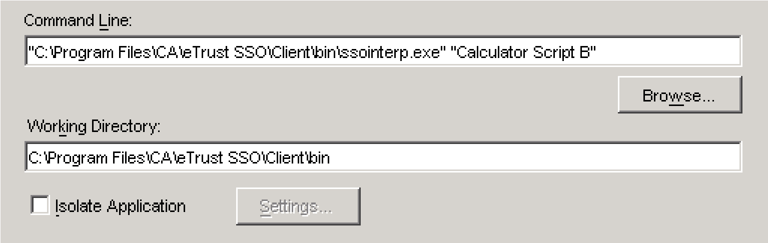
   The Publish Application dialog appears.

3. In the Publish Application dialog, chose the following:

   - Application (option button)
   - Command Line: Browse for ssointerp.exe in the CA SSO Client installation path then type the exact name of the application that has Script B assigned to it (you defined this using the Policy Manager)
   - Working Directory: This is the folder in which the ssointerp.exe is stored. This will be populated automatically if you browse for ssointerp.exe.

   For example:

```
"C:\Program Files\CA\Single Sign-On\Client\bin\ssointerp.exe" "Calculator Script B"
```



Command Line: "C:\Program Files\CA\eTrust SSO\Client\bin\ssointerp.exe" "Calculator Script B"

Working Directory: C:\Program Files\CA\eTrust SSO\Client\bin

☐ Isolate Application

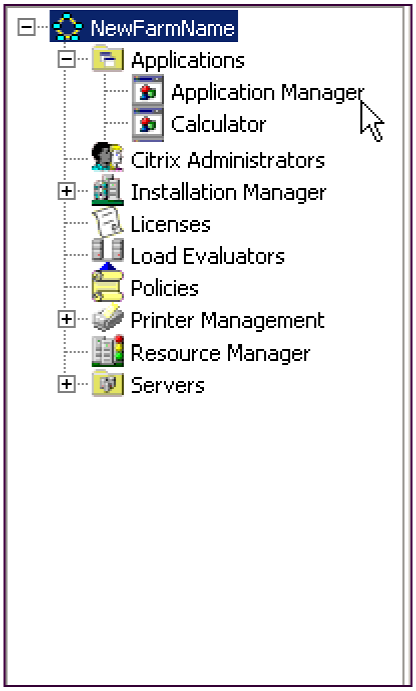
When you click Next the Program Neighborhood Settings dialog appears.

4. Continue through the Publish Application screens until you get to the Specify Servers dialog appears (you can accept the default information for all intervening screens).
5. Add all servers that should be able to run the published application.

   When you click Next the Specify Users dialog appears.
6. Add all Windows users or user groups that need access to this application.

   Note: *Do not* choose Allow Anonymous Connections, because application migration only supports explicit applications.

   When you click Finish you return to the Citrix Management Console.
7. Check that the application that you just published is visible in the Applications folder. You should see the Display Name that you entered in step 3.



Write Script A

This procedure tells you how to write an SSO script A. Script A runs on the ICA Client computer and launches the Citrix published application connection. Every application that you want SSO users to migrate must have its own Script A. It runs the Windows Script Host (WScript.exe) and passes the script name to it.

This procedure uses an example script and you can customize this to suit your environment.

To write a script A

1. Open a text editor and write Script A. You need the name of this script later when you are making your ICA Client connection.

   You can use this example and save it as Calculator_Script_A.tcl

   ```
   sso run -path WScript.exe -args "sessionConnect.js //Nologo Calculator $_LOGINNAME $_PASSWORD CitrixServer"
   ```

   sessionConnect.js

   JavaScript script name

   Calculator

   Citrix published application name

   CitrixServer

   Citrix MetaFrame Presentation server name on which the Citrix XTE Server is installed and running. It is also the system on which the SSO-enabled application is installed locally.

   LOGINNAME and PASSWORD

   These variables hold the credentials that the user uses to log on to the Citrix MetaFrame Presentation Server with. Each SSO user must have unique credentials to log onto the Citrix MetaFrame Presentation Server.

2. Save the Script A in the Scripts directory on the CA SSO Server.

   By default, the Scripts directory is found at:

   \Program Files\CA\Single Sign-On\Server\Scripts\

Write Script B

This procedure tells you how to write a CA SSO script B. Script B runs on the MetaFrame Presentation Server and launches the SSO-enabled Citrix published application. This script represents standard CA SSO functionality, but it is defined as a hidden application on the CA SSO Server. Every application that you want CA SSO users to migrate must have its own Script B.

This procedure uses an example script, but you could customize this to suit your environment. This is a simple example script that does not require a username and password. Most CA SSO-enabled applications would require a username and password.

To write script B

1. Open a text editor and write Script B in Tcl.

   You can use this example and save it as Calculator Script B.tcl.

   ```
   sso run -path {C:\WINDIWS\system32\calc.exe}
   ```

2. Save the Script B in the Scripts directory on the CA SSO Server.

   By default, the Scripts directory is found at:

   \Program Files\CA\Single Sign-On\Server\Scripts\

Define Script A on the CA SSO Server

This procedure describes how to define a Script A on the CA SSO Server. The Script A connects to the Citrix MetaFrame Presentation Server and tries to run the Citrix Published Application on the server.

To define Script A on the CA SSO Server

1. Launch the Policy Manager.
2. In the Program Bar navigate to Single Sign-On Resources, Application Resources, Application.
3. Right-click in the Application Window and choose New.

   The Create New APPL Resource – General dialog appears.

4. Fill in the details of the application.

   For example:

Name:	Calculator Script A
Caption:	Calculator
Type:	Desktop Application

   Note: The caption is what the user sees in their CA SSO Application List.

5. Click the Scripting button.

   The Scripting dialog appears.

6. Enter the Script A name in the Script File field, and then click OK.

   For example, Calculator_Script_A.tcl.

7. Select the Authorize icon.

   The Create New APPL Recourse – Authorize dialog appears.

8. Right-click and choose Add.

   The Add Access Control List Accessor dialog appears.

9. Choose the users who can access to this application, and then click OK.

   These should be the same users that you allocate to have access to Script B.

Define Script B on the CA SSO Server

This procedure describes how to define a Script B on the CA SSO Server. The Script B must be defined as a hidden application. This script launches the SSO-enabled application on the Citrix Server.

To define Script B on the CA SSO Server

1. Launch the Policy Manager

2. In the Program Bar navigate to Single Sign-On Resources, Application Resources, Applications.

   Right-click in the Application Window and choose New.

   The Create New APPL Resource – General dialog appears.

3. Fill in the details of the application.

   For example:

Name:	Calculator Script B
Caption:	Calculator_hidden
Type:	Desktop Application

4. Click the Scripting button.

   The Scripting dialog appears.

5. Enter the Script B name in the Script File field and click OK.

   For example: calculator_Script_B.tcl.

6. Click the Attributes icon.

   The View or Set APPL Properties – Attributes dialog appears.

7. Choose the Hidden checkbox.

8. Select the Authorize icon.

   The Create New APPL Recourse – Authorize dialog appears.

9. Right-click and choose Add.

   The Add Access Control List Accessor dialog appears.

10. Choose the users who can access to this application and click OK when you are finished. These users should be the same users that you allocated access to Script A.

Define the Application Credentials for Each User

This procedure tells you how to define the logon credentials for the SSO user to log on to the:

- Citrix MetaFrame Presentation server (Script A)
- SSO-enabled published application (Script B)

To define application credentials for a test user

1. Launch the Policy Manager and navigate to Single Sign-On Users, and find the test user.

2. Right-click the test user and choose Properties.

   The View or Set User Properties – General dialog appears.

3. Choose the Application List icon

   The View or Set User Properties – Application List dialog appears.

4. Choose the Script A application and click the Update Login Information button.

   The Update Login Information dialog appears.

5. Enter the users Windows credentials for login name and password for this user on the domain that permits the user access the published application on the Citrix MetaFrame Presentation server then click OK.

   Remember that this is the script that launches the published application link on the ICA client machine, so these credentials are what the user normally enters to logon to the Citrix MetaFrame Presentation server to access the application.

   Note: Every SSO user must have their unique logon details to the Citrix MetaFrame Presentation Server so that SSO can recognize individual sessions.

   Choose the Script B application and click the Update Login Information button.

   The Update Login Information dialog appears.

6. Enter the appropriate username (login name) and password for this user for the published application that runs on the MetaFrame Presentation Citrix server then click OK

   Note: In the example Script B for the Calculator, we do not make reference to a username and password, because Calculator does not have a logon screen. You would normally need to specify a username and password for the application that runs Script B. This Update Login Information dialog is where you enter the username and password that would be inserted into Script B.

How to Configure Closure of Previous SSO Session

To configure the closure of a previous SSO session, you must perform the following steps:

1. Enable SSO Session Management on the CA SSO Server (using the Policy Manager).
2. Define a Session Profile to restrict the user to only one SSO session.
3. Define the logoff command in the Client.ini file.

Test Application Migration

This procedure tells you how to test application migration. This is the procedure that end-users follow.

To perform this procedure, you need to:

- Set up the SSO Logoff Command.
- Set up SSO Session Management.

To test application migration

1. Using the test user, log on and authenticate to SSO on the ICA client machine. This creates a current SSO token.
2. Choose the application from the list of SSO-enabled applications.

   For example, Calculator, if you have defined it.

   The scripts should now launch the application.

3. Enter some numbers into Calculator. Remember these numbers so that you can test that you are opening the same session on the new machine.
4. Using the test user, logon and authenticate to CA SSO on a second ICA client machine.
5. Launch Calculator from the list of CA SSO-enabled applications.

   You should notice the Calculator application session close on the first ICA Client machine and the same session of Calculator with the numbers that you entered in step three on the second ICA Client computer.

Troubleshooting

Here are some troubleshooting tips to help you if you cannot get Application Migration working.

- Ensure the logon credentials that you used to access the Citrix MetaFrame Presentation Server are valid and that the user has the relevant Citrix privileges to run the published application.
- Make sure that you have a current valid SSO token by logging on with the SSO user again.
- Check the Tcl logon scripts
- Check the application script names in the Policy Manager
- Check that the Citrix published application name matched the name that you entered in Script A.
- Upgrade to the latest SSO Certified ICA Client version in case there are unexplainable connectivity issues.

Application Migration Configuration

This section tells you about ways you can configure Application Migration and how it works with the CA SSO Client.

Suspend ICA Client Connections During SSO Logoff

When the CA SSO Client is installed on the ICA Client workstations, a Javascript script called disconnectGroup.js (there is also a Visual Basic equivalent of this script) is installed in the CA SSO Client directory. This script automatically converts all open ICA Client connections to the “disconnected” state on the Citrix MetaFrame Presentation server when executed after the user logs off CA SSO on that workstation.

If the same user then logs onto SSO on another ICA Client workstation and starts one of the disconnected applications, the previous instance of that application will be returned to the user.

For example, you might configure this in the Client.ini file.

```
[EventCommands]
SsoSignOff=WScript.exe "%SSOINSTALLDIR%\bin\disconnectGroup.js"
```

Shared Computers and Session Management

Application Migration functionality is often used in conjunction with CA SSO session management in a shared computer environment. If you give every user a session profile that limits them to one CA SSO session and automatically closes their previous CA SSO session then applications follow users from computer to computer using the disconnectGroup.js logoff script discussed in the previous section. There is also a Visual Basic version of the logoff script called disconnectGroup.vbs.

For automatic session migration, you must set up Session Management in conjunction with Citrix Application Migration. Each CA SSO user must be allowed only one CA SSO session. This way the SSO session on the first workstation will automatically terminate when the CA SSO user logs onto the second workstation.

Chapter 15: Implementing Password Synchronization Agents

The Windows Password Synchronization Agent (PSA) lets you keep passwords synchronized between Active Directory (user's domain credentials) and the CA SSO Server. This is ideal for keeping SSO-enabled applications that require Windows credentials synchronized with the user's Windows credentials.

This section contains the following topics:

About Password Synchronization Agents

The Windows Password Synchronization Agent (PSA) lets you keep passwords synchronized between Active Directory (user's domain credentials) and the SSO Server. This is ideal for keeping SSO-enabled applications that require Windows credentials synchronized with the user's Windows credentials.

Bi-Directional

The Windows PSA is bi-directional, and is comprised of two components:

- Active Directory to CA SSO Server synchronization (password filter)
- CA SSO Server to Active Directory synchronization (password exit)

Both components can be installed on the same or different machines depending on whether the CA SSO Server is installed on the Domain Controller (DC). Typically, the Active Directory to CA SSO Server synchronization component is installed on every domain controller (PDC/BDC). The CA SSO Server to Active Directory synchronization component is installed on every CA SSO Server.

Decide on a Method of Installation

This section explains each type of installation to help you choose which method you should use.

The Password Synchronization Agents (PSA) can be installed using:

Installation wizard

The installation wizard leads you through the various steps required for installing the PSA. Use this method to familiarize yourself with the installation options.

Silent installation

Using the command line, you can silently install the PSA. You can also use this method to push the installation to remote computers.

If you choose to do a silent install, you must specify the variables by either:

- Creating a response file
- Using command line parameters

Note: The Windows command line may limit the number of characters in a single command. For complex or detailed command line installations, you should consider using a response file.

Pre-Installation Checklist

Use this checklist to make sure you have all the information and software that you need to install the Windows Password Synchronization Agent (PSA):

- Ensure that all system requirements are met before you begin installing the Password Synchronization Agent. For a complete list of system requirements, see the *SSO Readme* file.
- Ensure that all necessary prerequisite components have been installed, including the CA SSO Server.
- Ensure you are logged in as an administrator before installing the PSA components.
- If you plan to install the CA SSO Server to Active Directory PSA component, ensure that you have the following information:
  - List of synchronized applications.
  - Name of the CA SSO Server user data store.
  - List of Domain Controller machines with Active Directory LDAP SSL ports, for example, DC001:636 and DC002:636.

  - Trust file for the Domain Controller's certificates.
  - Active Directory administrator username and password.
  - Active Directory PSA username and password.

    Note: If this PSA component is being installed on an CA SSO Server farm, the PSA user name and password specified during installation on the first server need to be re-used for all remaining servers.

  - If you are not using the Active Directory as your user data store, you must provide LDAP search criteria to uniquely identify users in Active Directory with whom the synchronization is to take place.

- If you plan to install the Active Directory to CA SSO Server PSA component, ensure that you have the following information:
  - The name of the synchronized application. Usually a hidden master or domain application.
  - List of CA SSO Server machines.
  - CA SSO Server administrator username and password.
  - Name of the CA SSO Server user data store.
  - Active Directory to CA SSO Server user data store mapping filter, for example (sAMAccountName=%s).
- Ensure that the computer you are installing the PSA on has a TCP/IP connection with the CA SSO Server computers.
- You can choose not to use a Keystore when you install the PSA. This lets you install the PSA with a lower level of security. The Keystore is set in the WinPSAExit.ini file if you want to add it post-installation. If you supply a Keystore, the domain controller trust is verified by the agent. If it is not supplied, the agent to domain controller communications are encrypted but not authenticated.

Install the Windows PSA

This section explains how to install the Windows Password Synchronization Agent.

Install Using the Wizard

You should use this method to install the PSA on individual computers.

To install using the wizard

1. Insert the product installation DVD into your DVD drive.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Password Synchronization Agent, Windows Password Synchronization Agent.

3. Click Install and follow the prompts.

   Note: If you require more information, review the pre-installation checklist and introductory topics at the start of the section.

Install Using Silent Installation

You can install the Password Synchronization Agent (PSA) silently. This means that you need to provide the information that would normally be supplied by the administrator during graphical installation. This is done using the installation command (setup.exe, or equivalent) along with custom parameters set in the command line.

Before completing a silent install you must first read and accept the license agreement. You can do this by reading the EULA.txt file located at the root of the DVD, or alternatively, by using the installation wizard. You can find the command line setting required for accepting the license agreement and silently installing the PSA at the bottom of the license agreement.

Note: Windows command line (cmd.exe) imposes a 256 character limit in a single command. For complex or detailed command line installations, consider using a response file.

To install using silent installation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Windows Password Synchronization Agent.

3. Read the license agreement and note the command line setting for accepting the license agreement located at the bottom of the screen.

   You can now install the PSA using silent installation.

4. Open a command prompt and navigate to the PSA folder on the product DVD (<DVD>\password_sync\Windows_PSA).

5. From the command prompt, type:

   ```
   setup.exe -silent -V LICENSE_VIEWED=value {parameters}
   ```

   -silent

   Specifies a silent install.

   -V LICENSE_VIEWED=value

   Specifies whether you have viewed the license agreement found in the product install wizard or EULA.txt file.

   parameters

   Specifies the options to include in the silent install.

   For more information on command line options, see the next topic.

setup Command—Install Windows PSA

The command line parameters for installing the Password Synchronization Agent (PSA) include the following options:

-P installLocation

Defines the install location.

The command has the following format:

```
-P installLocation=[Value]
```

Value: The path to the install location surrounded by quotation marks if the path contains spaces.

-silent

Specifies a silent install.

The command has the following format:

```
-silent
```

-V IS_REBOOT_NOW

Specifies a system reboot once installation is completed.

The command has the following format:

```
-V IS_REBOOT_NOW=[Value]
```

Value: true | false

Default: false. Set to true if you want to allow the installer to reboot the system after installation, if it determines that a reboot is necessary.

-V LICENSED_VIEWED

Specify the product license key.

The command has the following format:

```
-V LICENSE_VIEWED=[Value]
```

Value: The value listed at the end of the license agreement on the product install wizard.

[Yes | No]

Active Directory to CA SSO Server component (Password Filter)

-P ad_to_ps_components.active

Specify whether to install the Active Directory to SSO Server synchronization components.

```
-P ad_to_ps_components.active=[Value]
```

Value: true | false

Default: True

-V ExistingAdminUser

Defines the name of the administrator user on the SSO Server.

```
-V ExistingAdminUser=[Value]
```

Value: Name of the administrator.

-V PSAdminPassword

Specify the password for the existing administrative user.

```
-V PSAdminPassword=[Value]
```

Value: The password for the administrative user.

-V PwdFilterSyncApp

Defines the name of the synchronization application defined on the SSO Server.

```
-V PwdFilterSyncApp=[Value]
```

Value: Name of synchronization application.

-V SearchFilter

Specify the LDAP search filter to be used if the user data store is not Active Directory.

```
-V SearchFilter=[Value]
```

Value: The LDAP search filter value for non-Active Directory data stores.

-V UserDataStore

Defines the name of the data store on the SSO Server. The data store contains the users to be used for synchronization.

```
-V UserDataStore=[Value]
```

Value: Name of the data store.

-V CommMode

Defines the mode of communication.

The command has the following format:

```
-V CommMode=[Value]
```

Value: 0 | 1 | 2

- 0 - Non-FIPS mode
- 1 - FIPS-only mode
- 2 - Mixed mode

-V IdentityFile

Defines the Trusted Certificate file in .pem format.

The command has the following format:

```
-V IdentityFile=[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

-V PrivateKeyPath

Defines the private key file in .pem format.

The command has the following format:

```
-V PrivateKeyPath =[FilePath]
```

FilePath: Indicates the absolute path to the file (along with file name)

CA SSO Server to Active Directory component (Password Exit)

-P ps_to_ad_components.active

Specify whether to install the SSO Server to Active Directory synchronization components.

```
-P ps_to_ad_components.active=[Value]
```

Value: true | false

Default: True

-V DomainControllers

Specify the list of domain controller hostnames/IP addresses and SSL port numbers. You can use a comma separated list to specify multiple domain controllers. The format is <name/IP>:<port.

```
-V DomainControllers=[Value]
```

Value: The list of domain controller hostnames/IP addresses and SSL port numbers.

-V Keystore

Specifies the location of the certificate trust file for the Domain Controllers.

-V Keystore=[*Value*]

Value: Location of the certificate trust file. The trust file typically has a .pem extension.

-V ADAdminUser

Specifies the full DN of the administrative user in the domain corresponding to the DomainControllers.

-V ADAdminUser=[Value]

Value: DN of the administrative user.

-V ADAdminPassword

Define the password for the Active Directory administrator user.

-V ADAdminPassword=[Value]

Value: The password.

-V ADExitUserDN

Specify the full DN of the user that is created in Active Directory for use in the synchronization process by the Password Exit.

-V ADExitUserDN=[*Value*]

Value: DN name.

-V ADExitUserGroup

Specify the full DN of the group which has sufficient permissions to reset domain user passwords, to which ADExitUserDN will be added to.

-V ADExitUserGroup=[*Value*]

Value: DN of the group.

-V ADExitUserPass

Specify the password for ADExitUserDN.

-V ADExitUserPass=[*Value*]

Value: Password.

-V SyncAppls

List the synchronization applications to be used by the SSO Server password exit. Use a comma-separated list to specify multiple applications.

-V SyncAppls=[*Value*]

Value: Names of the synchronization applications.

-V UserDataStore

Defines the name of the data store on the SSO Server. The data store contains the users to be used for synchronization.

```
-V UserDataStore=[Value]
```

Value: Name of the data store.

-V ExitUserStoreIsAD

Specify whether the user data store is of Active Directory type.

```
-V ExitUserStoreIsAD=[Value]
```

Value: 1 (True) | 0 (False)

-V ExitSearchBaseDN

Specify the base DN for the Active Directory user search that is performed when password synchronization takes place. Search criteria used involves ‘sAMAccountName’ attribute and a login name value for the configured synchronization application.

Note: Only applicable if the user data store is not of Active Directory type.

```
-V ExitSearchBaseDN=[Value]
```

Value: The base DN for the LDAP search, for example, CN=Users,DC=MyDomain,DC=COM

-V ExitSearchScope

Specify the type of scope to be used when performing Active Directory user search.

Note: Only applicable if the user data store is not of Active Directory type.

```
-V ExitSearchScope=[Value]
```

Value: The LDAP search scope, one of: Object, Subtree or One Level.

-V CommMode

Defines the mode of communication.

The command has the following format:

```
-V CommMode=[Value]
```

Value: 0 | 1 | 2.

- 0 - Non-FIPS mode
- 1 - FIPS-only mode
- 2 - Mixed mode

Install Using Silent Installation and Response File

Use the following procedure to install the Password Synchronization Agent (PSA) silently using a response file.

Note: You can use a combination of response file and command line options to silently install the PSA. In this case, the command line options override the response file. However, we recommend you use one method or the other to ensure there is no conflict in values.

Note: The following features might appear in the Password Synchronization Agent response file and should not be modified, as they are automatically generated and used internally by the InstallShield software:

- common_components.active
- win_common_components.active
- psexit_win32.active
- psexit_aix.active
- psexit_hp11.active
- psexit_hp23.active
- psexit_solaris.active
- psexit_linux.active

To install using silent installation and response file

1. Create a response file.
2. Open a command prompt and navigate to the PSA folder on the product DVD.
3. From the command prompt, type:

   setup.exe -silent [*parameters*] -options {*response file*}

   -silent

   Specifies a silent install.

   parameters

   Specifies the options to include in the silent install.

   -options response file

   Specifies that the InstallShield should use a response file. Also defines the name and location of the response file, for example c:\temp\ssorspfile.txt.

More information:

Create a Response File (see page 426)
setup Command—Install Windows PSA (see page 419)

Create a Response File

Response files record user specific install information and are commonly used in silent installations. Response files can be created using the command line *setup.exe -options-record* and specifying a file name. The *setup.exe* command launches the wizard install process where all information entered is recorded to the response file for reuse.

To create a response file

1. Open a command prompt and navigate to the PSA folder on the product DVD.
2. From the command prompt, enter:

   ```
   setup.exe -options-record {file name}
   ```

   -options-record file name

   Specifies that the InstallShield should generate a response file. Also defines the name and location of the generated file, for example c:\temp\ssorspfile.txt.

3. Complete the installation.

   The PSA is installed on the current machine.   In addition, a response file is created in the directory specified.

Post-Installation Requirements

The topics that follow describes required post-installation tasks.

Define a Computer as a Sessionless Terminal

If Session Management is required in your CA SSO system and you have installed the PSA, you must exempt the PSA computer.

To define a computer as a sessionless terminal

1. In the Policy Manager select Single Sign-On Resources, Configuration Resources, SSO Server Settings, Session Management.

   The View or Set GPSCONFIGURATIONPROPERTY Properties – Settings dialog appears

2. Double-click SessionlessTerminals.

3. The Edit Property dialog appears.
4. Enter the name of the computer the PSA is installed on.
5. Click OK.

Chapter 16: Scheduling Maintenance Tasks

You should set up a series of maintenance tasks for your CA SSO implementation.

To ensure your CA SSO implementation runs at an optimum level, you should regularly perform:

- CA SSO Server installation maintenance using the PSMaint utility
- Application list cache maintenance using the PSBGC utility
- CA SSO Server data backup

This section contains the following topics:

CA SSO Server Installation Maintenance

CA SSO provides a maintenance script to help you maintain and optimize your CA SSO Server installation. We recommend that you schedule some tasks to run regularly to keep your CA SSO installation well-tuned. The Server Maintenance scripts can be configured to perform the following tasks:

- Stop the services (CA SSO Server, CA Directory, CA Access Control)
- Remove any processed CA Access Control updates from the update file
- Restart the services (CA SSO Server, CA Directory, CA Access Control)
- Archive the logs for the specified service.

Schedule the Server Maintenance Tasks

To run the maintenance script regularly, use the following method.

To schedule regular server maintenance on Windows

1. From the Start menu, select Control Panel, Scheduled Tasks
2. Select Add Scheduled Task
3. Follow the wizard prompts, browse and select the PSMaint.cmd file and click Finish.

   The PSMaint.cmd is stored in the CA SSO Server utility directory. By default this is c:\Program Files\CA\Single Sign-On\Server\Utils.
4. Double-click the new scheduled task, and select the Task tab.
5. In the Run field, append the appropriate parameters outside the quotation marks and save.

   Note: You can also use the Windows “at” command. To learn more, open a command windows and type: at ?

PSMaint - Perform Server Maintenance

Use the PSMaint utility to schedule and run maintenance tasks on the CA SSO Server. You can use a number of different options with the maintenance utility. Each option lets you perform a different task or configure how that task will be run.

The following section outlines the options available with the utility. To run SSO maintenance tasks using the PSMAINT utility, you must be logged on as an administrator on Windows.

Windows syntax:

PSMaint.cmd *parameters*

-help

(Optional) Displays the help.

-stop | -start | -restart ***service(s)***

(Optional) Stops, starts or restarts the services specified. For Windows, separate multiple services with a dash ("-"). The services are indicated by one or more of the following values:

Note: We highly recommend that you use the "ALL" command to specify all services when starting and stopping, because there are dependencies between the services.

ALL

(Optional) All of the CA SSO services (CA SSO Server, CA Access Control, CA Directory).

PS or SSO

(Optional) CA SSO Server.

If you start the CA SSO Server (PS), CA Access Control and CA Directory (DIR) are also started if they are not running. These are dependent services.

DIR

(Optional) CA Directory DSAs.

If you stop or restart the CA Directory DSAs, you must also specify that the CA SSO Server should be likewise stopped or restarted. The CA SSO Server is a service that depends on the CA Directory DSAs.

AC

(Optional) CA Access Control (SEOS database). If you stop or restart CA Access Control, you must also specify that the CA SSO Server should be likewise stopped or restarted. The CA SSO Server is a service that depends on CA Access Control.

-clean_pmdb

(Optional) Removes processed CA Access Control PMDB updates in the updates file that have been replicated to other CA SSO Servers in a server farm. When CA Access Control is updated, a file called updates.dat stores the update so that the same update can be replicated. When the updates are replicated by default, the entry in the updates.dat file is not removed. To ensure that updates.dat file does not grow too large, you should use the –clean_pmdb flag to remove any replicated updates.dat files.

-pmdb name

(Optional) Specifies the PMDB name to use in the –clean_pmdb option. This overrides the auto-detected name, and is only required if a non-default pmdb name has been defined.

-seos path

(Optional) Specifies the path to CA Access Control (SEOS database). You would only need to use this if CA Access Control is not automatically detected and you get the error "The system cannot find the path specified" during the execution of an CA Access Control module.

-verbose | -v

(Optional)

On Windows, displays the command output but does not log it.

-log [logfile]

(Optional) Specifies a filename to log the command output to if you want to change the default. By default the system creates a PSM_Log.log file in the CA SSO Server log directory:

```
C:\Program Files\CA\Single Sign-On\Server\log\PSM_Log.log
```

-archive_logs *service(s) folder*

(Optional) Specifies whether to archive the log files for particular services. The services are listed below. This option creates a time-stamped .CAZ archive in the specified folder.

ALL

(Optional) All of the CA SSO services (CA SSO Server, CA Access Control, CA Directory).

PS or SSO

(Optional) CA SSO Server.

DIR

(Optional) CA Directory DSAs.

AC

(Optional) CA Access Control (SEOS database).

Example: Server maintenance on Windows

The following example shows you how to set the utility to perform the following tasks on a Windows platform:

- Run on the first day of every month at 11:30pm
- Stop all services
- Truncate CA Access Control updates file
- Restart all services

Note: This example assumes that you are logged on as an administrator.

```
PSMaint.cmd   –stop ALL –clean_pmdb –start ALL
```

Application List Cache Maintenance

For optimal performance the CA SSO Server accesses data from the application list cache instead of querying the CA Access Control repository.

A users application list cache can be updated in several ways:

- An administrator can update multiple user application list caches using the PSBGC utility
- An administrator can update a single user by logging on to the Policy Manager, opening that individual’s record, and selecting their Application List
- An end-user can update their own application list, by selecting the Refresh button on one of the Client Interfaces, for example, the SSO Tools window. This operation also updates the application list cache on the server side

Update the Application List Cache

Run the PSBGC utility to keep the application list cache up-to-date.

To update users' application list caches

1. Open a command line and navigate to the location of the PSBGC utility (by default in the CA SSO Server's bin directory).
2. Enter the following command.

   For Windows:

   ```
   psbgc –a [administrator name] –p [administrator password] parameters
   ```

psbgc - Update Application List Cache

Use the psbgc to keep users' applications lists up-to-date on the CA SSO Server. Specifically this command will update the application list cache. You can use a number of different options with the psbgc utility. Each option lets you perform a different task or configure how that task runs.

Windows syntax:

```
psbgc -a [administrator name] -p [administrator password] parameters
```

-h/-help

(Optional) Displays the help.

-a/-administrator

Specifies the administrator user name. This user must have administrative rights to the CA SSO Server.

When you install the CA SSO Server a psbgc utility administrator is created by default. This user is called “ps-bgc” and the corresponding password is “ps-bgc”. You must change this password to be more secure.

Note: When you run any psbgc utility command, you must specify an administrator and corresponding password.

-p/-password

Specifies the administrator's password ("ps-bgc" by default).

-c/-container

(Optional) Specifies the LDAP container name where the users are stored. This lets you calculate the application list for a specific subset of users within this container rather than all users in the directory. If this parameter is not specified, the psbgc updates all users within the user data store's base container (base path).

-d/-datastore

(Optional) Specifies the user data store where the users are stored. This lets you calculate the application list for all users within this data store. If not specified psbgc operates on all data stores.

-g/-group

(Optional) Specifies the name of the group of users for whom you want to calculate an application list.

-u/-user

(Optional) Specifies a single user for when you want to calculate a single user's application list.

-r/-recursive

(Optional) Specifies that the psbgc utility must calculate application lists recursively for all users within a specified container. This must be used in conjunction with the -c option if the specified container holds sub-containers that you also wish to search.

This means that you can use the psbgc utility to update all users' application lists within a hierarchy.

-x

(Optional) Specifies that the psbgc must use the paging technique and only return 200 users at a time. We highly recommended this is if you have more than 200 users to update and you are using a data store that supports paging.

-i/-ini

(Optional) Specifies the path to psbgc.ini configuration file.

By default this is stored in the CA SSO Server's bin directory on the CA SSO Server computer. If you are in the bin directory you do not need to specify where the psbgc.ini file is located.

Examples:

The following example shows how to perform the following tasks:

- Log onto the CA SSO Server (this example uses the default administrator name "ps-bgc" and password "password")
- Calculate and cache application lists for all users in all data stores
- Return results in batches of 200

```
psbgc -a ps-bgc -p password -x
```

- Calculate the application list for one user

```
psbgc -a ps-bgc -p password -d ps-ldap -c "ou=QA" -u "John Smith"
```

- Calculate the application list for all users under a specified container in an Active Directory data store

```
psbgc -a ps-bgc -p password -d AD -c "ou=QA"
```

CA SSO Server Data Backup

The topics that follow describe tasks for backing up the CA SSO Server data.

Back Up the CA SSO Server Configuration Data

You should maintain copies of all configuration files and scripts in a central location. You should also maintain backups of these files so that you can always re-implement the previous version if necessary.

The configuration files you should back up are:

Client configuration files

The Client.ini and Auth.ini files are located on each end-user computer, or in a central location if you choose to implement centralized CA SSO Client configuration.

Default location: \Program Files\CA\Single Sign-On\Client\cfg

MOTD files

If you use MOTD (message of the day) files, you should back these up.

Default location: \Program Files\CA\Single Sign-On\Server\Motd

Logon scripts

You should back up all logon and other scripts.

Default location: \Program Files\CA\Single Sign-On\Server\Scripts

Introduction to CA Directory Terminology

This section explains the CA Directory terminology used in this document:

DSA

A DSA is a process that manages some or all directory namespaces.

DSA console

The *DSA console* lets you connect to a DSA to give DXserver commands, receive trace information, and act as a user agent.

DXtools

The *DXtools* are a set of command-line utilities that come with CA Directory. These tools help you manage directory administration, work with LDIF data, load and unload data to and from a directory, and to extract and convert schemas for use with CA Directory.

How to Use the DXtools

You can run the DXtools in the following ways:

- Run the DXtools commands on the host, using the DSA console.
- Run the DXtools commands on a remote host, using the DSA console over a TCP/IP network.
- Include the DXtools commands in your scripts.

All tools return zero on success and non-zero when an error occurs.

DXHOME Environment Variable

Some tools require that the DXHOME environment variable is set to the home path of DXserver. This is done automatically when CA Directory is installed.

Some tools expect the DSA configuration files to be located in the *config* folder under the path in DXHOME.

Exit Status Codes for the DXtools

The DXtools share common exit codes, though not all exit codes apply to all tools. The exit codes are as follows:

0

Success

1

The corresponding DSA is running.

2

One or more of the datastore files already exists.

3

The specified directory location either does not exist or is not a directory.

4

The specified file is the wrong type, for example is a directory.

5

There is a permissions problem with this file.

6

The full path name of the datastore file is too large. This may be because the location specified for the datastore directory is too long.

7

An error occurred when trying to remove the old datastore files.

8

An error occurred when trying to rename the old datastore files.

9

An error occurred when trying to create or pad one of the files.

10

The datastore size is less than or equal to zero.

11

There was not enough space on the device or no memory available when trying to create the file.

12

There was insufficient access (perhaps because permissions were insufficient) to create the file or to set the access on the file.

13

The DXHOME environment variable is not set.

14

The DXHOME environment variable is not valid.

15

The corresponding DSA already exists.

16

The created DSA failed to start. Check its log files for details.

17

Incorrect or unknown command line parameters were provided.

18

The corresponding DSA does not exist.

How to Back up CA Directory Data

Use the following process to back up CA Directory data:

1. Connect to a local DSA console.
2. Take a snapshot copy of the datastore of the running default DSA. This process is named an online dump. Use the following command to take the snapshot:

   ```
   dump dxgrid-db;
   ```

Connect to a Local DSA Console

You can connect to a DSA locally on UNIX or Windows if a console port has been set for that DSA.

To connect to a local DSA Console

1. Open a command prompt on the host on which the DSA is running.
2. Enter the following command:

   ```
   telnet localhost local-port-number
   ```

   local-port-number

   Specifies the console port number of the DSA to which you want to connect.

Back up CA Directory Data

To back up CA Directory data

1. Connect to a local PS DSA console using the following port number:

   ```
   13379
   ```

2. Enter the following command:

   ```
   dump dxgrid-db;
   ```

   The data is backed up to the \dxserver\data folder.

   Note: You might not receive a notification that the backup is successful. Verify that a backup file with the extension .zdb is created with the same name and size of the original .db database file in the same directory to confirm that the backup is succesful.

3. Enter the following command to exit the console.

   ```
   logout;
   ```

   The CA Directory data is backed up.

More Information:

dump dxgrid-db Command—Take a Consistent Snapshot Copy of a Datastore (see page 442)

Schedule Back Up of CA Directory Data

To schedule back up of CA Directory data

1. Open the *.dxc file, associated with the database that you want to back up, from the following location:

   ```
   \dxserver\config\settings
   ```

2. Add the following commands to the *.dxc file associated with the database that you want to back up.

   ```
   dump dxgrid-db period <start> <period>;
   ```

   Where

   start

   Specifies the schedule time of the first backup. The value mentioned must be the time elapsed in seconds since Sunday 00:00:00 AM Greenwich Mean Time (GMT) of the week when you are scheduling the backup. For example, if you are scheduling a backup to run at 1 am on Sunday and then every twelve hours, specify the command as follows:

   dump dxgrid-db period 3600 43200;

   period

   Specifies the time interval in seconds between backups.

3. Run the following command to restart all directory services:

   ```
   dxserver init all;
   ```

   The back ups are scheduled.

More Information:

dump dxgrid-db Command—Take a Consistent Snapshot Copy of a Datastore (see page 442)

dump dxgrid-db Command—Take a Consistent Snapshot Copy of a Datastore

The *dump dxgrid-db* command takes a consistent snapshot copy of the datastore of a running DSA (an online dump). The DSA completes any updates before carrying out this command and does not start any more updates until the copy is finished.

The datastore files are copied to files with extensions starting .z:

- A database file *.zdb
- An attributes file *.zat
- An object classes file *.zoc

Note: Each dump overwrites the previous backup files. If you want to save the backup files, copy them to another location before the next dump.

The DXdumpdb tool can export data from a datastore created by the dump command.

The command has the following format:

```
dump dxgrid-db [period start period];
```

period *start period*

(Optional) Specifies that the online dump is performed at regular intervals.

start

Defines the number of seconds since Sunday 00:00:00 am GMT.

Note: The start time is defined using GMT and not your local time.

period

Defines the number of seconds between online dumps.

Example: Perform an Online Dump Every Hour

The following command takes a snapshot copy of the database files and the DSA information every 12 hours:

```
dump dxgrid-db 0 43200;
```

On a peer data DSA server, you schedule a different start time as follows:

```
dump dxgrid-db 3600 43200;
```

Note: Ensure that you create a cron job on UNIX or a scheduled task on Windows to copy the backed up files to a safe location. Each dump overwrites the previous backup files.

Restore CA Directory Data

To restore CA Directory data

1. Run the following command from command prompt to shut down all directory services:

   ```
   dxserver stop all
   ```

   Note: Also, stop all CA SSO services.

2. Rename or delete the following files associated with the corrupted DSA:

   - *.db
   - *.at
   - *.oc

   For example, if the DSA PS_ServerName is corrupted, delete the PS_ServerName.db, PS_ServerName.at, and PS_ServerName.oc files.

3. Rename the backed up files as follows:

   - *.zdb file to *.db file
   - *.zat file to *.at file
   - *.zoc file to *.oc file

   For example if PS_ServerName.zdb is the backed up database file, rename it to PS_ServerName.db. Similarly, rename the .zat and .zoc files.

4. Run the following command from a command prompt to start all directory services:

   ```
   dxserver start all
   ```

   The back ups are restored.

Back Up the SSO Resource Data

The CA Access Control data store (seosdb) stores all resource data, including application and authentication hosts.

This data does not change frequently, but you should back it up regularly.

Depending on how often the resource information changes, you could back up the resource data store nightly or weekly.

See the CA Access Control documentation for further information on Backup and Restore options. You can download these manuals from Support Online.

To back up the resource data store

1. By default, you will find the dbmgr tools on the CA SSO Server computer in the CA Access Control bin directory. Open a command line and navigate to the CA Access Control bin directory. By default, under Windows this is:

   ```
   C:\Program Files\CA\Access Control\bin
   ```

2. Enter the following command:

   ```
   dbmgr -backup foldername
   ```

   The resource data store is backed up to the folder you specified.

To restore the resource data store

1. Open a command line and navigate to the CA Access Control bin directory.
2. Enter the following command to stop CA Access Control:

   ```
   secons –s
   ```

   The CA Access Control data store stops.

3. Copy the backup files from the backup directory to the seosdb directory.

   If you have a server farm configuration, also copy the backup file to the pmdb data directory.

4. Enter the following command to start CA Access Control under Windows:

   ```
   seosd –start
   ```

   The CA Access Control data store starts.

Example: Restore a Resource Data Store

The following example shows you how to restore CA Access Control on a Windows computer where:

- The previous backup of the resource data store is in this directory: C:\Backup\01012006
- The current resource data store is in this directory: C:\Program Files\CA\Access Control\data\

1. Open a command line.
2. Stop CA Access Control using the following command:

   ```
   secons –s
   ```

3. Copy the backup files from the backup directory to the seosdb and pmdb directories using the following two commands:

   ```
   copy /Y "C:\Backup\01012006\seosdb\*.*" "C:\Program Files\CA\Access Control\data\seosdb"
   ```

   ```
   copy /Y "C:\Backup\01012006\pmdb\*.*" "C:\Program Files\CA\Access Control\data\pmdb"
   ```

   The second command here, for the pmdb, only applies to a server farm environment.

4. Enter the following command to start CA Access Control:

   ```
   seosd –start
   ```

   The CA Access Control data store starts.

How to Synchronize CA Directory Databases

Use the following process to synchronize CA Directory databases when adding a new server to the server farm, or during disaster recovery. Synchronizing databases ensures that all the databases contain the same information.

1. Back up CA Directory Data from a working or existing CA SSO Server.
2. Use the backup files from the working CA SSO server and synchronize these files with the databases on the new or corrupted CA SSO Server.

More Information:

Back up CA Directory Data (see page 440)

Synchronize Databases of New or Corrupted CA SSO Server

When a new server is being added to a server farm or a corrupted CA SSO server is being restored, synchronize the associated databases with a database of an existing and working CA SSO server.

To synchronize databases

1. Copy the following backed up files of the existing or working database of an CA SSO server to the new or corrupted server
   - *.zdb
   - *.zat
   - *.zoc
2. Shutdown the directory services and CA SSO services on the new or corrupted server.
3. Rename or delete the .db, .at, and .oc files of the corrupted server.
4. Rename the .zdb, .zat, and .zoc files copied from the working server to reflect the names of the deleted .db, .at, and .oc files on the new or corrupted server.

   For example, if you are copying the backup files of Server1 to Server2. Rename the PS_Server1.zdb, PS_Server1.zat, and PS_Server1.zoc files to PS_Server2.db, PS_Server2.at, and PS_Server2.oc files respectively.
5. Run the following command from a command prompt to start the directory services.

   ```
   dxserver start all
   ```

   The databases are synchronized.

Chapter 17: Upgrading

This section contains the following topics:

Upgrade CA SSO

This section provides information on upgrading CA SSO from r8 (CA SSO Client components only) and r8.1 to r12. It includes information on:

- Upgrading all CA SSO components
- Upgrading the CA SSO Server
- Backward compatibility
- Upgrading CA SSO data stores (data migration)

The following sections explain what a typical upgrade process from CA SSO r8.1 to r12 looks like. It includes the following:

- Upgrading CA SSO Servers from r8.1 to r12
- Upgrading Authentication Agents from r8.1 to r12
- Upgrading CA SSO Clients from r8.1 to r12

CA SSO Server Data Migration Upgrade Paths

Data Migration paths can be used to upgrade existing CA SSO r8 or CA SSO r8.1 to r12 in the following situations:

- CA SSO Server is installed on the same machine; however use of the upgrade wizard is not possible, for instance in CA SSO r8 to r12 scenario.
- CA SSO Server r12 is installed on a separate machine from existing r8.0 or r8.1 Server and the existing configuration needs to be migrated to a new machine.

With some minor differences the data migration procedure in both cases would involve the following:

- Exporting data from the user and the policy data stores of the existing CA SSO r8.0 or r8.1 Server.
- Installing fresh CA SSO r12 Servers.
- Processing data taken on step 1 with one or more Migration Tools.
- Loading resulting data into CA SSO r12 Server.

Upgrade from r8 to r12

CA SSO does not support a direct upgrade from r8 to r12. You can either:

- Set up a new CA SSO r12 environment. This involves:
  - Installing r12 Servers.
  - Migrating all data from r8 to r12.
- Upgrade to r8.1 Servers and then to r12 Servers.

Upgrade from r8.1 to r12

CA SSO Server supports the automatic upgrade from r8.1 to r12. You can either:

- Upgrade directly to r12. This involves:
  - Running the r12 installation wizard and upgrading all components
  - Upgrading all users to CA SSO Client r12
- Maintain backward compatibility. This involves:
  - Upgrading r8.1 CA SSO Servers to r12
  - Upgrading r8.1 Authentication Agents to r12
  - Running existing r8.1 Clients against r12 Servers and then upgrading users to the CA SSO Client r12 over time

Note: If CA SSO Servers r8.1 use embedded Ingres 2.6, the r12 Server (this is the case for most of 8.1 servers that have been upgraded from CA SSO 8.0) Installation Wizard is not able to upgrade them. In this case, Ingres 2.6 needs to be upgraded to Ingres 3.0 manually using the following procedure:

1. Download the latest 8.1 CR Server from CA Support Online and unpack the installation package
2. Back up the entire system.
3. Dump the data to an LDIF file using the DXdumpdb tool:

   ```
   Dxdumpdb -f c:\ps.ldif (DSNAME of PS on the machine)
   ```

4. Sort the LDIF data using the ldifsort tool:

   ```
   ldifsort c:\ps.ldif c:\ps_sorted.ldif
   ```

5. Stop CA SSO Server and CA Directory services from the Services applet
6. Destroy old Ingres 2.6 database using the DXdestroydb tool:

   ```
   dxdestroydb ps
   ```

7. Navigate to the location where new CA SSO Server r8.1 installation package is located. Go to CA Directory installation directory. For instance if your CA SSO Server r8.1 package resides in C:\sso81server, issue the following command:

   ```
   cd c:\ sso81server\eTrustDirectory\dxserver\windows
   ```

8. Upgrade CA Directory. This removes Ingres 2.6 and install Ingres r3.

   ```
   dxsetup ETRDIR_DXSERVER_EMBEDDED=1 CALLER_ID=ETSSO ETRDIR_SHORTCUTS=0
   ```

   Follow instructions on the screen. At the beginning of the upgrade process the installer asks you to stop Ingres service and verify that none of the Ingres components are running. Wait until installation completes with "Installation completed successfully" dialog.

9. Create new PS database using the DXnewdb tool:

   ```
   dxnewdb ps
   ```

10. Reload the data from the LDIF files using the DXloaddb tool:

    ```
    Dxloaddb (DSANAME of PS on the machine) ps.ldif
    ```

11. Start Directory Services, CA SSO Server Service.
12. Verify CA SSO Server is operational.

Now you can run CA SSO r12 Server Installation Wizard, which upgrades the server to r12.

CA Access Control User Data Store Migration

If CA Access Control is used as a primary user data store in CA SSO r8.0, the upgrade to r12 will require migration of this user data store to either embedded CA Directory (ps-ldap) or Active Directory Data Store.

In the case of Active Directory the migration is a two stage process. The first stage involves migrating data to the CA SSO r12 platform using the ps-ldap data store. The second stage involves migrating the data to an Active Directory data store.

Backward Compatibility with CA SSO Clients

r12 Servers support r8.1 Clients working in backward compatibility mode. In this scenario, you can upgrade your CA SSO Servers to r12 and continue to run CA SSO Clients r8.1 components without any modifications.

When you are ready to start upgrading r8.1 Clients to r12, upgrade r8.1 authentication agents for all deployed authentication methods to r12 first.

r8.1 Clients working in backward compatibility mode can authenticate against r8.1 or r12 agents, while r12 Clients must authenticate against r12 agents. This may require adjusting server sets definitions for the r12 Clients.

Authentication Agents Backward Compatibility with CA SSO Clients

r12 Authentication Agents support backward compatibility with r8.1 Clients.

About Data Store Migration

All upgrades from r8 to r12 require data migration to the CA SSO Server r12 platform. Upgrade from r8.1 to r12 can also be done using data migration scenario.

The migration is a two stage process. The first stage involves migrating data to the r12 platform using the ps-ldap data store. The second stage involves migrating the data to an Active Directory data store.

The different types of data include:

- User logon information (logininfos) - stored in the PS DSA
- User data - stored in PS DSA or Active Directory
- Administrative data - stored in Access Control
- Token data (not replicated or synchronized)

All other information can be replicated and synchronized using CA Access Control, CA Directory and (if appropriate) Active Directory.

Note: With an Active Directory data store, all user information is in Active Directory. However, the users' application login information (LoginInfo objects) is in ps-ldap (CA Directory).

More information:

Migrate SSO Data Stores (see page 455)

Upgrade Administration Tools

This section explains how to upgrade each of the management tools.

Upgrade Policy Manager

The Policy Manager is a Windows application that should be installed on each administrator's workstation. The Policy Manager is an important tool for configuring CA SSO.

To upgrade Policy Manager

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Policy Manager.

   The Policy Manager installation wizard appears.

3. Select Install and follow the prompts.

More information:

Implementing the Policy Manager (see page 117)

selang

Selang remains the command line language used for managing the CA Access Control data base. For more information about selang, see the *CA Access Control selang Reference Guide.*

Upgrade Session Administrator

The upgrade process for Session Administrator is as follows:

- Un-install previously installed version of Session Administrator

More information:

Install the SSO Session Administrator (see page 370)

Upgrade the CA SSO Client

When you upgrade to CA SSO Client r12, all existing configuration settings (stored in SSOCInt.ini) are mapped to two new configuration files:

- **Client.ini** - contains all the core functionality and interface settings
- **Auth.ini** - contains all information about authentication and server sets

Note: If you are installing CA SSO Client r12, ensure you have installed the r12 versions of the CA SSO Server r12 and authentication agents. CA SSO Client r12 is not backward compatible with earlier CA SSO versions.

To upgrade the CA SSO Client on one machine

1. Insert the product DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer main menu, select Single Sign-On Client r12.
3. Select the Upgrade option.
4. Follow the prompts to the ServerSet Configuration Migration dialog.

   Select Yes if you want to migrate your existing ServerSet information to the CA SSO r12 Client. This option requires the pre-installation of version r12 of the CA SSO Server and authentication agents.

   Select No if you *do not* want to keep your existing ServerSet information, or if you have not yet upgraded your CA SSO Server or authentication agents to version r12.

5. Check that the CA SSO Client is working correctly. You may need to manually edit the Client.ini or Auth.ini files.

Upgrade the CA SSO Server

You can use the Product Explorer to automatically upgrade existing CA SSO Servers. Direct upgrades of earlier CA SSO r8 are not supported. For r8 you can:

- Create an r12 environment and migrate existing data to the new platform.
- Upgrade to r8.1 Servers and then to r12 Servers.

This section explains how to upgrade the CA SSO Server from r8.1 to version r12.

Important! When upgrading or doing any maintenance of the CA SSO Server, you must use the same user account under which the product was initially installed.

Pre-Upgrade Requirements

Be aware of the following requirement before you upgrade to CA SSO r12:

If the Policy Manager is installed on the same machine as the CA SSO Server, it must be removed before upgrading the server.

If CA SSO Server r8.1 uses Ingres 2.6, it should be upgraded to Ingres 3.0 first.

More Information

Upgrade from r8.1 to r12 (see page 448)

Upgrade Other Server Side Components

This section contains procedures for upgrading other Server side components.

Upgrade Windows Password Synchronization Agent (PSA)

You can upgrade PSA to r12 by running the PSA r12 install option available on the CA SSO r12 Product Explorer.

The CA SSO r12 PSA includes bidirectional components:

- Active Directory to CA SSO Server password synchronization
- CA SSO Server to Active Directory password synchronization

To upgrade the PSA (Windows)

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO r12 PSA.

   The authentication agent installation wizard appears.

3. Select Install.

   The Installation Wizard detects that you have an old version and will uninstall it before proceeding.

4. Once the old version is uninstalled, follow the prompts and click Install.

Upgrade Authentication Agents

CA SSO supports the following authentication agents:

- Certificate (Windows)
- LDAP (Windows)
- RSA (Windows)
- WIN (Windows)

You can upgrade to the current version of each agent by running the relevant install option available on the Product Explorer. This install process detects earlier versions of the software and uninstalls old versions before installing the new version.

To upgrade an authentication agent (Windows)

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select the relevant authentication agent.

   The authentication agent installation wizard appears.

3. Select Install.

   The Installation Wizard detects that you have an old version and will un-install it before proceeding.

4. Once the old version is un-installed, follow the prompts and click Install.

Migrate SSO Data Stores

CA SSO supports the following data stores:

- CA Directory in r8, r8.1 and r12
- Active Directory in r8, r8.1 and r12

The following section guides you through migrating user data from CA Directory to Active Directory.

Note: Even with migrating all data to Active Directory, SSO CA Directory is still retained in SSO r12 and is used to store user login and application information. All other user data is stored in Active Directory.

Important! The CA SSO Server data migration tools do not support migration of data which contains characters other than those of the English locale from previous versions of CA SSO to r12.

Data Migration Process

Typically, the data migration process consists of the following steps:

1. Export the CA SSO Server’s data from the existing CA Access Control and CA Directory data stores.
2. Export CA SSO Server configuration settings (registry settings or ini files)
3. Move the data collected in Step 1 and 2 to the new CA SSO r12 platform. Run data migration tools to convert the data to the new format supported by r12 servers.
4. Load the embedded CA Directory (CA SSO r12) with the modified data.
5. Load the embedded CA Access Control (CA SSO r12) with the modified data.
6. Migrate the data to Active Directory.

Pre-Upgrade Requirements

Use this checklist to make sure you are aware of all requirements before you migrate CA SSO data from r8 to r12 and Active Directory:

- Ensure the CA SSO r12 machine has Java 2 JRE 1.4.x. Make sure the CLASSPATH includes a path to the CA SSO Server Utils directory which contains the migration tools. The default location is: c:\Program Files\CA\Single Sign-On\Server\Utils.

- All CA SSO r12 user information is managed by Active Directory, however, SSO application login information remains in the CA Directory ps-ldap data store, and access rules are still stored in CA Access Control (the administrative directory in your r12 implementation).
- ADMigrate.exe is installed by default when installing the CA SSO Server on Windows.

  Note: If ADMigrate.exe is run on a Windows machine without CA SSO Server installed, you need to first install the VC 2005 C++ redistributable package, which is available from the Third Party section of the CA SSO installer DVD.

Migrate from CA SSO r8 or r8.1 to r12 (CA Directory or Active Directory as a User Data Store)

This section takes you through migrating the CA SSO r8.0 or 8.1 data to r12 when CA Directory or Active Directory is used as a primary data store.

1a. Exporting Data from SSO 8.0 or 8.1

This procedure takes you through collecting CA SSO r8.0/8.1 data and copying it to the CA SSO r12 target machine.

1. Export the data from CA Access Controll using the following command:

   ```
   dbmgr -e -r -f {output selang file}
   ```

   where

   -e

   Specifies the export mode

   -r

   Specifies whether to use the database being used by the CA Access Control engine. To use this option, make sure that the CA Access Control engine is running.

   -f {output selang file}

   Specifies the data output file and directory.

2. Export the ps-ldap data using the dxdumpdb command, for example:

   ```
   Dxdumpdb -f ps.ldif (DSANAME of PS on the machine)
   ```

3. Locate all SSO application specific data, including TCL script files.

   The TCL script files on a Windows system are typically found in "c:\Program Files\eTrust\SSO\Server\scripts". Get all files in this directory.

4. Copy all information collected in Steps 1-3 to the target CA SSO r12 machine.

1b. Process the Data

To process the data, you must open a command prompt on the SSO r12 machine and issue the following command;

```
java MigrateResources -in {selang file from step 1}
```

This creates the file MigrateResources.selang which contains selang commands for resources and access control lists to be loaded into CA Access Control.

1c. Load the Data

This procedure takes you through loading the data.

To load the processed data

1. Copy the tcl scripts to the SSO r12 Server scripts directly.
2. From the command prompt, type:

   ```
   selang -f MigrateResources.selang
   ```

   The resources are loaded into CA Access Control.

The next step involves loading the users and LoginInfo into the target CA Directory. You can do this using either the dxmodify or dxloaddb command. These commands are CA Directory utilities located in the CA Directory dxserver/bin directory, and are documented in the CA Directory Reference Guide.

Load the Data Using dxmodify

To load the data using dxmodify

1. Prepare MigrateUsers.ldif for loading by processing it through ldifsort. ldifsort ensures that each record is followed by its immediate subordinates, and removes duplicate entries, for error-free loading. From the command prompt, type:

   ```
   ldifsort MigrateUsers.ldif users.ldif
   ```

2. Load the users and LoginInfo information into the target CA Directory data store (ps-ldap) using dxmodify.

   ```
   dxmodify –a -c –h localhost:13389 –D “cn=ldap-admin,o=ps” –w "{administrator password}" –f users.ldif
   ```

   Note: This command may generate errors regarding entries that already exist. This is to be expected. However, if there are errors other than "Already exists" errors, it would be worth investigating.

If MigrateUsers.ldif is not empty, load its sorted file using a similar command as above.

All users and login information is loaded into CA Directory.

-c will continue the loading even if errors are encountered. Errors will be encountered because the sorted MigrateUsers.ldif file will contain entries that are duplicates of existing entries.

It is also recommended that you backup the CA Directory data prior to performing dxmodify. Dxdumpdb can be used for this.

Load the Data Using dxloaddb

To load the data using dxmodify

1. Extract the data from ps-ldap into an ldif file using dxdumpdb. For example:

   ```
   Dxdumpdb -f ps.ldif (DSANAME of PS on the machine)
   ```

   Note: dxdumpdb is an CA Directory utility.

2. Concatenate the files ps.ldif, MigrateLoginInfo.ldif, and MigrateUsers.ldif if it is not empty. You can use any editor or the copy command. For example:

   ```
   copy ps.ldif + MigrateLoginInfo.ldif + MigrateUsers.ldif result.ldif
   ```

   Note: Check the resulting file and end-of-line characters.

3. Prepare the resulting ldif file for loading by processing it through ldifsort. For example:

   ```
   ldifsort result.ldif resultSorted.ldif
   ```

4. Shutdown the CA SSO Server and the CA SSO Server CA Directory DSAs.

5. Load the ldif file to CA Directory. For example:

   ```
   dxloaddb resultSorted.ldif ps
   ```

Migrate from SSO 8.0 to SSO 12 (Access Control as a User Data Store)

This section takes you through migrating the CA SSO r8.0 Access Control data store to CA SSO r12 (CA Directory or Active Directory data store).

The migration takes place in two stages. The first involves migrating the data to CA SSO r12 using the ps-ldap data store; and the second stage, which is optional, involves migrating the data to an Active Directory data store.

1. Migrate data to SSO r12 Using the ps-ldap Data Store

To migrate the CA SSO r8.0 data to r12

1. Export data from CA SSO r8.0
2. Process the data using the CA SSO migration tools
3. Load the processed data into the target CA SSO r12 system
4. Fix all data stores aside from ps-ldap that use ldap-pers as the administrator.

Note: This procedure needs to be repeated for all CA SSO r12 server farm members.

1a. Export the Data

This procedure takes you through collecting SSO r8.0 data and copying it to the SSO r12 target machine.

To export the data

1. Export the data from SSO r8.0 using the following command:

   ```
   dbmgr -e -r -f <output selang file>
   ```

   where

   -e

   Specifies the export mode

   -r

   Specifies whether to use the database being used by the Access Control engine. To use this option, make sure that the CA Access Control engine is running.

   -f {output selang file}

   Specifies the data output file and directory.

2. Export the data and login information from the CA SSO Server DSA to an ldif file. For example:

   ```
   Dxdumdb –f {ldif file} (DSANAME of PS on the machine)
   ```

   where

   -f {ldif file}

   Specifies the data output file and directory.

3. Locate all SSO application specific data, including TCL script files.

   The TCL script files on a Windows system are typically found in “c:\Program Files\eTrust\SSOServer\scripts”. Get all files in this directory.

4. Copy all information collected in Steps 1-3 to the target SSO r8.1 machine.

1b. Process the Data

This procedure takes you through processing the SSO r8.0 data using SSO's migration tools.

To process the data

1. On the SSO r12 machine, open a command prompt and type the following:

   ```
   java MigrateResources -in {selang file from procedure 1a} -ver 8.0
   ```

   This creates the file MigrateResources.selang which contains selang commands for resources and access control lists to be loaded into Access Control.

2. In the command prompt, type:

   ```
   java MigrateUsers -in {selang file from procedure 1a}
   ```

   This creates four files:

   - **MigrateUsers.ldif.** Contains user and logininfo entries (with appropriate attributes encoded in base 64) to be loaded to the target CA Directory datastore.
   - **MigrateUsers_NoBase64Encode.ldif.** Contains the same information as MigrateUsers.ldif but with none of the attributes encoded.
   - **MigrateUsers.selang.** Contains selang commands for creating administrative users (if any are migrated) and commands that remove users from groups.

     Note: Removing users from groups is useful if the target SSO r12 contains users and groups that may conflict with the data from SSO r8.0. If migrating to an SSO r12 fresh install, this is not needed, but if there are administrative users to migrate, this file must be loaded.
   - **MigrateUsers_DeleteUsers.selang.** Contains selang commands to delete users and groups. This is useful if the target SSO r12 contains users and groups that may conflict with the data from SSO r8.0. If migrating to an SSO r12 fresh install, this is not needed.

3. In the command line, run:

   ```
   java MigrateLoginInfo -in <ldif file from step 1a> -ver 8.0
   ```

   This creates 1 file:

   MigrateLoginInfo.ldif:

   Contains users and login info objects in r12 format. Check the usage of MigrateResources to change this name.

   Note: To see other options, such as specifying the name of the target CA Directory datastore, run java MigrateResources without any arguments.

1c. Load the Data

This procedure takes you through loading the data.

To load the processed data

1. Copy the tcl scripts to the SSO r12 Server scripts directory.
2. From the command prompt, type:

   ```
   selang -f MigrateResources.selang
   ```

   The resources are loaded into Access Control.
3. If required, and the files are not empty, type:

   ```
   selang -f MigrateUsers.selang
   ```
4. From the command prompt, type:

   ```
   selang -f MigrateUsers_DeleteUsers.selang.
   ```

   Note: On a fresh install, this may result in errors, which is to be expected, since the users and groups being deleted do not exist.

The next step involves loading the users and LoginInfo into the target CA Directory. You can do this using either the dxmodify or dxloaddb command. These commands are CA Directory utilities located in the CA Directory dxserver/bin directory, and are documented in the CA Directory Reference Guide.

For more information on loading the data using DXmodify, see Load the Data Using dxmodify (see page 462)

For more information on loading the data using DXloaddb, see Load the Data Using dxloaddb (see page 463)

Load the Data Using dxmodify

To load the data using dxmodify

1. Prepare MigrateUsers.ldif for loading by processing it through ldifsort. ldifsort ensures that each record is followed by its immediate subordinates, and removes duplicate entries, for error-free loading. From the command prompt, type:

   ```
   ldifsort MigrateUsers.ldif users.ldif
   ```

2. Load the users and LoginInfo information into the target CA Directory data store (ps-ldap) using dxmodify.

```
dxmodify -a -c -h localhost:13389 -D “cn=ldap-admin,o=ps” -w
"{administrator password}" -f users.ldif
```

Note: This command may generate errors regarding entries that already exist. This is to be expected. However, if there are errors other than "Already exists" errors, it would be worth investigating.

If MigrateUsers.ldif is not empty, load its sorted file using a similar command as above.

All users and login information is loaded into CA Directory.

-c continues the loading even if errors are encountered. Errors are encountered because the sorted MigrateUsers.ldif file contains entries that are duplicates of existing entries.

It is also recommended that you backup the CA Directory data prior to performing dxmodify. Dxdumpdb can be used for this.

Load the Data Using dxloaddb

To load the data using dxmodify

1. Extract the data from ps-ldap into an ldif file using dxdumpdb. For example:

   ```
   Dxdumdb –f ps.ldif (DSANAME of PS on the machine)
   ```

   Note: dxdumpdb is an CA Directory utility.

2. Concatenate the files ps.ldif, MigrateLoginInfo.ldif, and MigrateUsers.ldif if it is not empty. You can use any editor or the copy command. For example:

   ```
   copy ps.ldif + MigrateLoginInfo.ldif + MigrateUsers.ldif result.ldif
   ```

   Note: Check the resulting file and end-of-line characters.

3. Prepare the resulting ldif file for loading by processing it through ldifsort. For example:

   ```
   ldifsort result.ldif resultSorted.ldif
   ```

4. Shutdown the CA SSO Server and the CA SSO Server CA Directory DSAs.

5. Load the ldif file to CA Directory. For example:

   ```
   dxloaddb resultSorted.ldif ps
   ```

1d. Fix the Data

Fix the Data

1. If there are any data stores, other than ps-ldap, that use ldap-pers as the administrator, change its administrator to another user. This is because ldap-pers is an internal user that is intended for use by the CA SSO Server only. Any other user data store must define its own administrator.

2. Set up the passwords for EAC authentication for migrated administrative users in the Access Control data store.

2. Migrate the Data to Active Directory

This procedure takes you through migrating data from the SSO r12 ps-ldap user data store to the Active Directory (AD) data store.

Once migrated, all SSO r12 user information is managed by Active Directory, however, SSO application login information remains in the CA Directory ps-ldap data store. Thus, it is important that prior to performing these steps, user information as well as user groupings, are already defined in Active Directory.

The migration tool used is ADMigrate.exe. This tool is found in the CA SSO Server bin directory.

Migration to an Active Directory data store involves the following general steps:

1. Set up Active Directory data store.
2. Export the Active Directory data.
3. Export the ps-ldap data.
4. Process the data using the AD migration tool.
5. Load the resulting ldif file into ps-ldap.

If ADMigrate.exe is run on a Windows machine without CA SSO Server installed, you need to first install the VC 2005 C++ redistributable package, which is available from the Third Party section of the SSO installer DVD.

2a. Set Up Active Directory Data Store

Prior to migration, an Active Directory datastore must be set up. The following needs to be performed:

- Create and configure a data store. It is recommended that the LoginInfo Container DN be: "ou=<ad-datastore-name>,ou=LoginInfos,o=PS".
- Configure a dxlink router.
- Set up users, groups, and possibly organizationalUnit containers in the Active Directory.
- Set up authorization for usage of SSO resources by Active Directory user groups.
- Configure an LDAP authentication agent.
- Configure CA SSO Clients for LDAP authentication.

More information:

Active Directory as the User Data Store (see page 220)

2b. Export the Active Directory Data

To export the data

1. Open a command prompt.
2. Export the Active Directory data using ldifde or csvde . For example:

```
ldifde -a “cn=Administrator,cn=Users,dc=addomain,dc=com” password -s localhost -d
“cn=Users,dc=addomain,dc=com” -r “(objectClass=User)” -l sAMAccountName -m -f ad_ldif.txt
```

or

```
csvde -a “cn=Administrator,cn=Users,dc=addomain,dc=com” password -s localhost -d
“cn=Users,dc=addomain,dc=com” -r “(objectClass=User)” -l sAMAccountName -m -f ad_csv.txt
```

Note: The above commands are examples only and need to be modified for your environment.

2c. Export the ps-ldap Data

To export ps-ldap data using dxdumpdb

1. Open a command prompt.
2. Export the ps-ldap data using the dxdumpdb command, for example:

```
Dxdumdb –f ps.ldif (DSANAME of PS on the machine)
```

2d. Process the Data Using the AD Migration Tool

To process the data

1. Open a command prompt.
2. Export the Active Directory data, for example:

```
ADMigrate -ad_ldif ad_ldif.txt -ps_ldif ps_ldif.txt -ad_name ad-datastore -ad_base_path
"cn=Users,dc=addomain,dc=com"
```

Note: The above example command line uses default values for most of the arguments.

The ADMigrate.exe tool creates the following files:

- ADMigrate.ldif - which contains the converted LoginInfo information to be loaded to ps-ldap.
- ADMigrate_Unmatched.ldif - Contains the ps-ldap user and LoginInfo entries of non-migrated users.
- ADMigrate.log - Contains conversion information and error messages.

Note: You can use an INI file to record options and simplify the command line.

2e. Load the ldif File into ps-ldap

To load the ldif file into ps-ldap

1. Shut down the CA SSO Server.

   Note: If you are going to use dxloaddb, you need to stop the directory DSA's in addition to the CA SSO Server. To stop all DSAs, type:

   ```
   Dxserver stop all
   ```

2. Prepare the ADMigrate.ldif file for loading to ps-ldap by processing with ldifsort, for example, running the command:

   ```
   ldifsort ADMigrate.ldif sortedADMigrate.ldif
   ```

3. Load the resulting ldif file into ps-ldap using dxloaddb, for example:

   ```
   Dxloaddb PS sortedADMigrate.ldif
   ```

4. Start the CA SSO Server.

Migrate Users From CA Access Control to CA Directory

This section explains additional points what you need to know about migrating your user data from CA Access Control to CA Directory.

Prerequisites

- Ensure that you do not have any users and user groups with the same name.

  In the Access Control user data store it is possible for a user and a user group to have the same name. However in CA Directory, a group and a user cannot have the same name. The migration utility assumes that all users and groups have unique names. Therefore, before you begin this migration, you must replace any duplicate group/user names and rename the authorizations.

- Determine if there are any usernames that include brackets characters:"(" or ")".

  Usernames with brackets are not supported in LDAP and are not migrated with this utility. You must rename these users to remove the brackets.

- Determine the impact of the fact that users who are members of the default user groups in the Access Control data store, are not be members of those groups in CA Directory data store after migration.

  By default, when the CA SSO Server is installed, the default user groups are automatically created. When you run the migration utility, those users are migrated to CA Directory, but they no longer belong to those user groups.

  Here is a list of the default user groups that are created when the CA SSO Server is installed.

  - _abspath
  - _interactive
  - _network
  - _pr-adms
  - _restricted
  - _surrogate

What Is Not Migrated

The MigrateUsers utility deliberately does not migrate Access Control administrator users or any of the default users or groups which are listed here:

- ps-admin
- pswd-pers
- nobody
- RSV
- _undefined
- _seagent
- _abspath
- _interactive
- _network
- _ps-adms
- _restricted
- _surrogate

Migrate From SSO r8.1 ps-ldap User Data Store to an SSO r12 Active Directory User Data Store

This procedure takes you through migrating data from the SSO r8.1 ps-ldap user data store to the Active Directory (AD) data store.

Once migrated, all SSO r12 user information is managed by Active Directory, however, SSO application login information remains in the CA Directory ps-ldap data store. Thus, it is important that prior to performing these steps, user information as well as user groupings, are already defined in Active Directory.

The migration tool used is ADMigrate.exe. This tool is found in the SSO Server bin directory.

Migration to an Active Directory data store involves the following general steps:

1. Set up Active Directory data store.
2. Export the Active Directory data.

3. Export the ps-ldap data.
4. Process the data using the AD migration tool.
5. Load the resulting ldif file into ps-ldap.

Note: ADMigrate.exe is installed by default when installing the CA SSO Server on Windows.

If ADMigrate.exe is run on a Windows machine without CA SSO Server installed, you need to first install the VC 2005 C++ redistributable package, which is available from the Third Party section of the SSO installer DVD.

1. Set Up the Active Directory Data Store

Prior to migration, an Active Directory datastore must be set up. The following needs to be performed:

- Creation and configuration of a datastore. It is recommended that the LoginInfo Container DN be:
  "ou=<ad-datastore-name>,ou=LoginInfos,o=PS".
- Configuration of a dxlink router.
- Set up users, groups, and possibly organizationalUnit containers in the Active Directory.
- Set up authorization for usage of SSO resources by Active Directory user groups.
- Configuration of an LDAP authentication agent.
- Configuration of SSO Clients for LDAP authentication.

2. Export the Active Directory Data

To export the data

1. Open a command prompt.
2. Export the Active Directory data using ldifde or csvde . For example:

```
ldifde –a “cn=Administrator,cn=Users,dc=addomain,dc=com” password –s localhost –d
“cn=Users,dc=addomain,dc=com” –r “(objectClass=User)” –l sAMAccountName –m –f ad_ldif.txt
```

   or

```
csvde –a “cn=Administrator,cn=Users,dc=addomain,dc=com” password –s localhost –d
“cn=Users,dc=addomain,dc=com” –r “(objectClass=User)” –l sAMAccountName –m –f ad_csv.txt
```

   Note: The above commands are examples only and need to be modified for your environment.

3. Export the ps-ldap Data

To export ps-ldap data using dxdumpdb

1. Open a command prompt.
2. Export the ps-ldap data using the dxdumpdb command, for example:

   ```
   Dxdumdb –f ps.ldif (DSANAME of PS on the machine)
   ```

   Note: To export ps-ldap data on UNIX, type:

   ```
   su - dsa -c "/opt/CA/eTrustDirectory/dxserver/bin/dxdumpdb ps"
   ```

4. Process the Data Using the AD Migration tool

To process the data

1. Open a command prompt.
2. Export the Active Directory data, for example:

   ```
   ADMigrate -ad_ldif ad_ldif.txt -ps_ldif ps_ldif.txt -ad_name ad-datastore -ad_base_path
   "cn=Users,dc=addomain,dc=com"
   ```

   Note: The above example command line uses default values for most of the arguments.

   The ADMigrate.exe tool creates the following files:

   - ADMigrate.ldif - which contains the converted LoginInfo information to be loaded to ps-ldap.
   - ADMigrate_Unmatched.ldif - Contains the ps-ldap user and LoginInfo entries of non-migrated users.
   - ADMigrate.log - Contains conversion information and error messages.
   - **Note:** You can use an INI file to record options and simplify the command line.

5. Load the ldif File into ps-ldap

To load the ldif file into ps-ldap

1. Shut down the SSO Server.

   Note: If you are going to use dxloaddb, you need to stop the directory DSA's in addition to the SSO Server. To stop all DSAs, type:

   ```
   Dxserver stop all
   ```

2. Prepare the ADMigrate.ldif file for loading to ps-ldap by processing with ldifsort, for example, running the command:

   ```
   ldifsort ADMigrate.ldif sortedADMigrate.ldif
   ```

3. Load the resulting ldif file into ps-ldap using dxloaddb, for example:

   ```
   Dxloaddb PS sortedADMigrate.ldif
   ```

4. Start the SSO Server.

Chapter 18: Uninstalling

You can use the Product Explorer to install or uninstall any CA SSO component. In addition to this, you can use the Product Explorer to modify some of the components.

You can tell if a component is already installed because it appears in bold in the Product Explorer window.

This section contains the following topics:

Uninstall the CA SSO Client

You can choose to uninstall the CA SSO Client entirely, or just some components of the CA SSO Client, for example, the GINA pass through functionality.

Note: To uninstall the CA SSO Client on Vista when UAC is enabled you require elevated privileges.

To uninstall the CA SSO Client

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select the CA SSO Client option.

   The Uninstall button becomes active.

3. Click the Uninstall button and follow the prompts.

   The CA SSO Client is uninstalled.

4. Click Finish.

   You may be asked to restart the computer.

Uninstall CA SSO Client Components

This procedure tells you how to un-install the CA SSO Client components without uninstalling the CA SSO Client itself. The components that you can remove include:

- GINA Upgrade & Station Lock
- GINA Pass Through
- Citrix ICA Client support
- SSO Credential Provider

To uninstall CA SSO Client components

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select the CA SSO Client option.

   The Uninstall button becomes active.

3. Click Uninstall and follow the prompts ensuring you select the items you want uninstalled.

   The CA SSO Client components are uninstalled.

Uninstall Using Add/Remove Programs

Use the following procedure to remove the CA SSO Client on a Windows machine.

To uninstall the CA SSO Client or Client components

1. From the Windows Start Menu, select Control Panel, Add or Remove programs.
2. Select CA SSO Client and click Change/Remove.
3. Follow the prompts to remove this program or alternatively, components of the CA SSO Client.

Uninstall Using Unicenter Software Delivery (USD)

You can uninstall the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD).

Note: The following procedures assume you have USD installed and operational.

To uninstall using USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to uninstall.
3. Click the sub folder Installations.
4. Right-click All Computers and Users, Schedule Configure Job, Uninstall.

   Note: For more uninstall options, see the *Unicenter Software Delivery Online Help*.

Uninstall the CA SSO Server

The topics that follow describe how to uninstall the CA SSO Server.

Uninstall on Windows

Use the following procedure to remove the CA SSO Server on a Windows machine.

To uninstall the CA SSO Server

1. From the Windows Start Menu, select Control Panel, Add or Remove programs.
2. Select CA SSO Server and click the Change/Remove button.
3. Follow the prompts to remove the CA SSO Server.

Uninstall on Windows Using Command Line

Use the following procedure to remove the CA SSO Server on a Windows machine using the command line.

To uninstall the CA SSO Server

1. Open the command prompt.
2. Uninstall the CA SSO Server installation by typing:

   [*install location*]_uninstall\uninstaller.exe

   install location

   The CA SSO Server's install location.

   _uninstall\uninstaller.exe

   Uninstall the software.

3. Follow the prompts to remove the CA SSO Server.

   Note: Uninstalling the CA SSO Server will also uninstall the Policy Manager if it exists on this machine.

Uninstall the Policy Manager

This procedure tells you how to uninstall the Policy Manager.

Note: You can also use the Add/Remove programs utility located in the Control Panel.

To uninstall the Policy Manager

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select the Policy Manager for Windows option.

   The Uninstall button becomes active.

3. Click Uninstall.

   The Welcome screen appears.

4. Click Next.

   The Program Maintenance dialog appears.

5. Select the Remove option and click Next.

   The Remove the Program dialog appears.

6. Click Remove.

   The Policy Manager is uninstalled and the InstallShield Wizard Completed dialog appears.

7. Click Finish.

   The Policy Manager is now uninstalled.

Uninstall Using Add/Remove Programs

Use the following procedure to remove the Policy Manager on a Windows machine.

To uninstall the Policy Manager

1. From the Windows Start Menu, select Control Panel, Add or Remove programs.
2. Select Policy Manager and click Change/Remove.
3. Follow the prompts to remove this program.

Uninstall the Session Administrator

This procedure tells you how to uninstall the Session Administrator.

Note: You can also use the Add/Remove programs utility located in the Control Panel.

To uninstall the Session Administrator

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select the Session Administrator option.

   The Uninstall button becomes active.

3. Click Uninstall.

   The Welcome screen appears.

4. Click Next.

   The Program Maintenance dialog appears.

5. Select the Remove option and click Next.

   The Remove the Program dialog appears.

6. Click Remove.

   The Session Administrator is uninstalled and the InstallShield Wizard Completed dialog appears.

7. Click Finish.

   The Session Administrator is now uninstalled.

Uninstall Using Add/Remove Programs

Use the following procedure to remove the SSO Session Administrator on a Windows machine.

To uninstall the SSO Session Administrator

1. From the Windows Start Menu, select Control Panel, Add or Remove programs.
2. Select Session Administrator and click Change/Remove.
3. Follow the prompts to remove this program.

Uninstall Using Unicenter Software Delivery (USD)

You can uninstall the CA SSO Client and Session Administrator using Unicenter Software Delivery (USD).

Note: The following procedures assume you have USD installed and operational.

To uninstall using USD

1. Click Start, Programs, Computer Associates, Unicenter, Software Delivery, SD Explorer.
2. Click Software Library, All Software and the select the software install package you want to uninstall.
3. Click the sub folder Installations.
4. Right-click All Computers and Users, Schedule Configure Job, Uninstall.

   Note: For more uninstall options, see the *Unicenter Software Delivery Online Help*.

Uninstall an Authentication Agent

This procedure tells you how to uninstall an authentication agent.

Note: You can also use the Add/Remove programs utility located in the Control Panel.

To uninstall an authentication agent

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select the Authentication Agent you wish to uninstall.

   The Uninstall button becomes active.

3. Click Uninstall and follow the prompts.

   The Authentication Agent is uninstalled.

Uninstall the Password Synchronization Agent

The topics that follow describe how to uninstall the PSA.

Uninstall on Windows

Use the following procedure to remove the Password Synchronization Agent

To uninstall the Password Synchronization Agent

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.

2. Select CA SSO Password Synchronization Agent.

   The Uninstall button becomes active.

3. Click Uninstall and follow the prompts.

   After rebooting the machine, the Password Synchronization Agent is uninstalled.

Uninstall on Windows Using Command Line

Use the following procedure to remove the Password Synchronization Agent (PSA) on a Windows machine using the command line.

To uninstall the PSA

1. Open the command prompt.
2. Uninstall the PSA installation by typing:

   [*install location*]_uninst\uninstaller.exe

   install location

   The PSA's install location.

   _uninst\uninstaller.exe

   Uninstall the software.
3. Follow the prompts to remove the PSA.

Uninstall the Documentation

Use the following procedure to remove the documentation.

To uninstall the documentation

1. Insert the product installation DVD.

   If you have Autorun enabled, the Product Explorer automatically runs. Otherwise, navigate to the DVD drive and double-click the PE_i386.EXE file.
2. Select any of the items in the Documentation Folder.

   The Uninstall button becomes active.
3. Click Uninstall and follow the prompts.

   The CA SSO documentation is uninstalled.

Uninstall the SSO Application Wizard

To uninstall the SSO Application Wizard, use Add or Remove Programs in the Windows Control Panel.

To uninstall the SSO Application Wizard

1. From the Windows Start Menu, select Control Panel, Add or Remove programs.
2. Select CA SSO Application Wizard, and then click Remove.

   The SSO Application Wizard is uninstalled from the computer.

Chapter 19: Performing an Advanced Example Implementation

The following scenario is designed to help get you started with CA SSO as quickly as possible. It guides you through the steps to set up CA SSO in a specific configuration.

This section contains the following topics:

Configuration Scenario Outline: SSO with Active Directory

The following is a description of a typical CA SSO installation. This scenario does not require access to any live systems. For example, you do not need access to the domain controller to set up this test scenario. We suggest that you set up this scenario in a test environment before you install this in a live environment.

This is one of the most common CA SSO configurations. The key points of the scenario are:

Data Stores - User Information

Store users in Active Directory. In this example, the company has existing users that are stored in a hierarchical structure and they want to configure CA SSO to use the existing user repository.

Data Stores - Application Access Information

Store users in Active Directory. Users and/or groups will determine which applications each end-user has single sign-on access to.

Store application access information in CA Access Control.

Data Stores - Logon Information

Store logon information in CA Directory installed on the CA SSO Server (supplied with CA SSO).

Authentication

Use LDAP authentication against Active Directory data store.

Operating Systems You Will Need

To set up this scenario in a test environment you need the software listed below. The first column lists the suggested names of the machines; these are the names used in this chapter to refer to the computer.

During this test implementation you will set up a sample domain called 'AcmeCorp.com' - you may want to replace this with a domain naming scheme more relevant to your enterprise.

Machine Name	Operating System	SSO Components Installed
SSOserver1	Windows 2003 SP2 Server	CA SSO Server Directory Server
SSOserver2	Windows 2003 SP2 Server	CA SSO Server Policy Manager
SSOclient	Windows XP SP 2	CA SSO Client
SSOauthag	Windows 2003 SP2 Server	LDAP Authentication Agent. **Note:** This should be on a separate computer to the other components)

How to Implement the Scenario

The following process summarizes the steps to set up the example scenario for CA SSO. The rest of this chapter explains each step in detail. We recommend that you work through this chapter in the order.

1. Configure SSOserver1.AcmeCorp.com as a domain controller for AcmeCorp.com with Active Directory.
2. Create a number of test users within Active Directory.
3. Install the CA SSO Servers in a server farm setup on SSOserver1 and SSOserver2.
4. Install the Policy Manager on SSOserver2.
5. Configure the CA SSO Server:
   a. Create a DSA router to Active Directory
   b. Configure the Directory Access Controls to allow the DXlink to Active Directory.
   c. Use Policy Manager to create a user data store on the CA SSO Server to use Active Directory.
   d. Use the Policy Manager to create a new LDAP authentication host record.
   e. Verify user data store configuration.
   f. Authorize SSO resources to Active Directory user groups.
   g. Apply Resources.
6. Install and configure the LDAP Authentication Agent on SSOauthag.
7. Install and configure the CA SSO Client on SSOclient.
8. Use LDAP to authenticate to Active Directory from SSOclient.
9. Create and test an application:
   a. Create a logon script.
   b. Define the logon script on the CA SSO Server.
   c. Launch the application.

Step 1: Configure Windows 2003 As A Domain Controller

To set up a Windows 2003 Server as a Domain Controller

Note: You may need the Windows 2003 installation CD during the setup.

1. From the Start menu on your Windows 2003 Server, select Programs, Administrative Tools, Configure Your Server Wizard.

   The Windows 2003 Configure Your Server Wizard dialog appears.

2. Click Next twice.

   The Server Role screen shows the server roles that have been added to the server.

3. If the Domain Controller (Active Directory) role has not been configured, select this role from the list and click Next..
4. Scroll down and select the Start link.

   The Active Directory Installation Wizard appears.

5. Click Next, then select “Domain Controller for a new domain” option.
6. Follow the prompts to configure the Domain Controller. You can accept the defaults. When prompted to enter the New Domain Name (full DNS), enter the demo domain name: AcmeCorp.com
7. When you are finished, restart the computer as prompted.

Step 2: Create Test Users in Active Directory

To set up some test users in Active Directory in a hierarchical structure

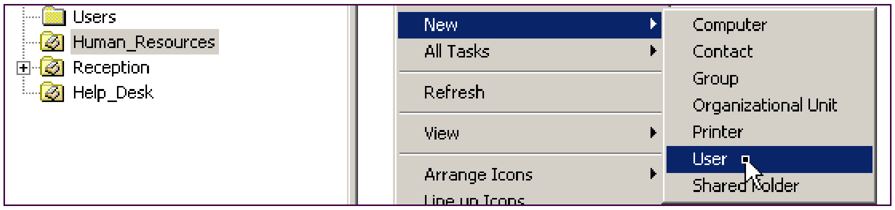
Note: The examples in this procedure are referred to throughout this scenario.

1. Log on to SSOserver1.AcmeCorp.com as a Windows user with administrative privileges (i.e. is a member of the ‘Administrators’ group), preferably as a built-in ‘Administrator’ account.
2. From the Start menu select, Programs, Administrative Tools, Active Directory Users and Computers.

   The Active Directory Users and Computers dialog appears.

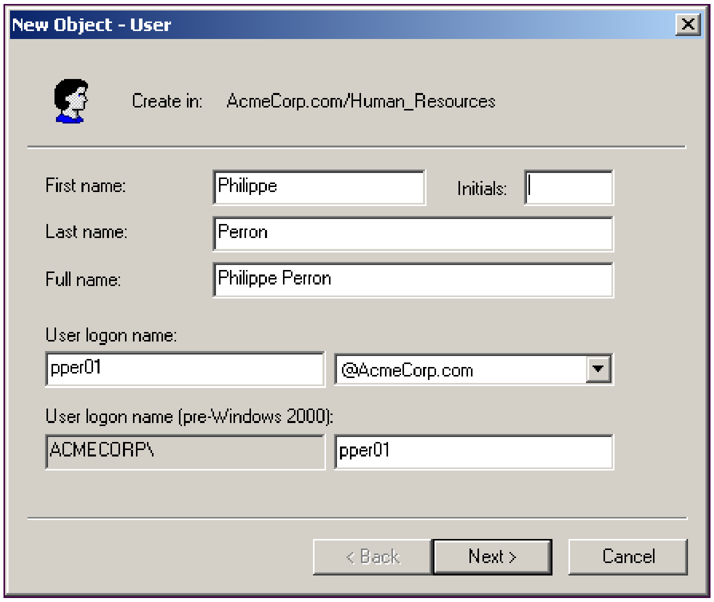
3. Select the domain (AcmeCorp.com) in the left pane, then right-click in the right pane and select New, Organizational Unit from the menu.
4. Create three new organizational unit folders:
   - Human_Resources
   - Help_Desk
   - Reception

5. Select the Human_Resources folder from the tree in the left pane, then right-click in the right pane and select New, User from the menu.



The New Object – User dialog appears.

6. Fill in the necessary fields to create a test user called Philippe Perron, then click Next.
First name: Philippe
Last name: Perron
User logon name: pper01



New Object - User

Create in: AcmeCorp.com/Human_Resources

First name: Philippe

Initials:

Last name: Perron

Full name: Philippe Perron

User logon name:

pper01 @AcmeCorp.com

User logon name (pre-Windows 2000):

ACMECORP\ pper01

7. Enter and confirm the user password, leaving the checkboxes empty. Click finish to create the user object in Active Directory.

   Remember the password. You will use it later in the chapter.

8. Repeat steps 5, 6 and 7 to create two other test users:

   Organizational unit: Help_Desk
   First name: Prani
   Last name: Patil
   User logon name: ppat01

   Organizational group: Reception
   First name: Penelope
   Last name: Price
   User logon name: ppri01

9. Select the Users folder from the tree in the left pane, then right-click in the right pane and select New, Group from the menu.

10. Enter the Group name as ssoUsers and click OK.

11. For each of the three users created, right click on the user name and select 'Add members to a group' from the menu. Add each user to the ssoUsers group.

Step 3: Install the CA SSO Server Farm

For information about installing the CA SSO Server in a server farm configuration, see the "Implementing the CA SSO Server" chapter of this guide.

More information:

Implementing the CA SSO Server (see page 91)

Step 4: Install the Policy Manager

To install the Policy Manager

1. Insert the product installation DVD into your DVD-ROM drive.

   If you have Autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the DVD-ROM drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select Configuration Tools, Policy Manager, and click Install.

3. Follow the wizard prompts and when you are done, click Install to start the Policy Manager installation.

   You can accept all the default settings when you install the Policy Manager.

   Note: If you previously installed the CA SSO Server on this computer, you need to stop CA Access Control when prompted to do so.

Step 5: Configure the CA SSO Server

Step 5a: Create a DSA Router to Active Directory

On the SSOserver1 machine where you have set up Active Directory, create a router DSA to connect the CA SSO Server to Active Directory. This router uses an CA Directory DSA to route LDAP traffic from the local CA Directory server to the Active Directory server (using a DXlink).

To create a DXLink between CA Directory and Active Directory

1. Using Windows Explorer, go to the following directory:

   ```
   C:\Program Files\CA\Directory\dxserver\config\knowledge
   ```

2. Create an empty text file named Router_AD.dxc. Substitute ACMECORP for the actual AD domain name. This file creates a router DSA called Router_AD, and points to the Active Directory on SSOserver1.

   Note: If Windows Explorer is set to hide extensions, the file may incorrectly be created with the extension ".dxc.txt". This is not correct and you must change the Windows Explorer setup and rename with just the extension .dxc.

3. Using notepad, edit the Router_AD.dxc. Make sure that you have SSOServer1 as the machine host name, and the domain is AcmeCorp.com as shown in the example below:

   - "svrpol01" to the Active Directory computer name
   - "acmecorp" to your domain name

Note: The domain components in the last parameter are displayed in reverse order; this is how the DSA expects it.

```
# Computer Associates DXserver/config/knowledge
# Router_AD.dxc
# Routes to Active Directory on ACMECORP domain
# Refer to the Admin Guide for the format of the set dsa command.
set dsa Router_AD =
{
    prefix           = <dc "com"><dc "AcmeCorp">
    native-prefix    = <dc "com"><dc "AcmeCorp">
    dsa-name         = <o AD_ACMECORP><cn Router_ADr>
    dsa-password     = "secret"
    address          = tcp "SSOServer1" port 389
    auth-levels      = clear-password, ssl-auth
    dsa-flags        = read-only
    trust-flags      = allow-check-password, no-server-credentials
    link-flags       = dsp-ldap, ms-ad
};
set transparent-routing = true ;
```

Note: Please note the following points that affect this file:

- In the address line, make sure that the host name (computer name) of the Domain Controller is there: SSOserver1.
- The "read-only" dsa-flag prevents updates to AD from the CA SSO Server (even if the account used by the user data store has domain admin privileges). This is intentional: the CA SSO Server sees Active Directory as a source of user data, but as user management is outside of the scope of SSO, this should happen directly within Active Directory using an AD management console or similar.

4. Using notepad, open PS_Servers.dxg and add the name of the file above to end the file.

   ```
   source "..knowledge/Router_AD.dxc";
   ```

   For example:

   ```
   # Computer Associates DXserver/config/knowledge/

   #

   # PS_Servers.dxg written by CA CA SSO Server Installation
   #
   # Description:
   # Use this file to group and share DSA knowledge.
   # PS DSA's source this file
   # from its initialization file.
   #
   source "../knowledge/PS_SSOserver1.dxc";
   source "../knowledge/PSTD_SSOserver1.dxc";
   source "../knowledge/Router_AD.dxc";
   ```

   You must now restart the eDIR and CA SSO Server and CA Directory services.

5. Go to the Windows Start menu and select Programs, Administrative Tools, Services.
6. Stop the CA Directory and CA SSO Server services.
7. Find the services called CA Directory - PS_CA SSO Server1 and CA SSO Server.
8. Right-click and select Start from the menu. This will also load the CA Directory Services.

Step 5b: Configure the Directory Access Controls to allow the DXlink to Active Directory

To update the Directory Access Controls to create a dxlink to the Active Directory data store.

1. Using Windows Explorer, go to the following directory:

   ```
   C:\Program Files\CA\Directory\dxserver\config\access
   ```

2. Open the PS_Access.dxc file in a text editor.

3. Add the following information to the group section at the top of the file:

   ```
   set group = {
   name = "AD_Group"
   users = <dc "com"><dc "acmecorp"><ou "Help_Desk"><cn "Prani Patil">
   };
   ```

   This adds the user ‘Prani Patil’ from the Active Directory data store to a group name ‘AD_Group’

4. In the “Give Admin users access to PS and PSTD tree's” section add the following:

   ```
   set admin-user = {
   group = "AD_Group"
   subtree = <dc "com"><dc "acmecorp">
   };
   set admin-user = {
     group = "AD_Group"
     subtree = <o "PS"><ou "LoginInfos"><ou “ad-acmecorp”>
   };
   ```

   This configures the directory to allow a connection to read the Active Directory tree as long as the user trying to access it through the CA SSO Server DSA is listed in the AD_Group Access Controls group. This also allows access to the portion of CA SSO Server’s ps-ldap DSA where application login information objects are stored.

Step 5c: Create a User Data Store on the CA SSO Server to use Active Directory

To create a user data store that points to AD for user records and local LDAP for the user’s login information

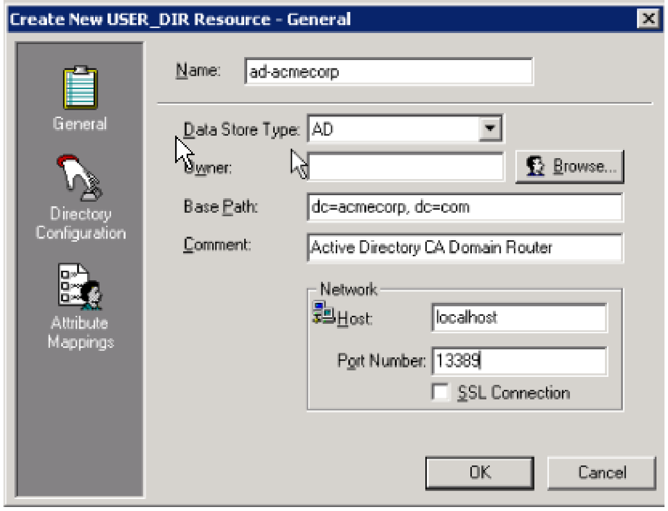
1. Log onto the Policy Manager.
2. Go to Resources, Single Sign-On Resources, User Resources, Datastores.
3. Right-click in the right pane and select New from the menu.
4. Enter the following in the dialog:
   - Name: ad-acmecorp
   - Data Store Type: AD
   - Owner: [blank]
   - Base Path: dc=acmecorp, dc=com
   - Comment: Active Directory CA Domain Router
   - Host: localhost
   - Port: 13389

5. Click the Directory Configuration icon on left.

6. Configure the datastore using the following dialog. You should use a permanent user, but they do not need to be an administrator. For example,

   Admin: cn=Prani Patil, ou=Help_Desk, dc=acmecorp, dc=com

   Password: whatever you assigned to this user when creating it.



Create New USER_DIR Resource - General

Name: ad-acmecorp

Data Store Type: AD

Owner:

Base Path: dc=acmecorp, dc=com

Comment: Active Directory CA Domain Router

Network

Host: localhost

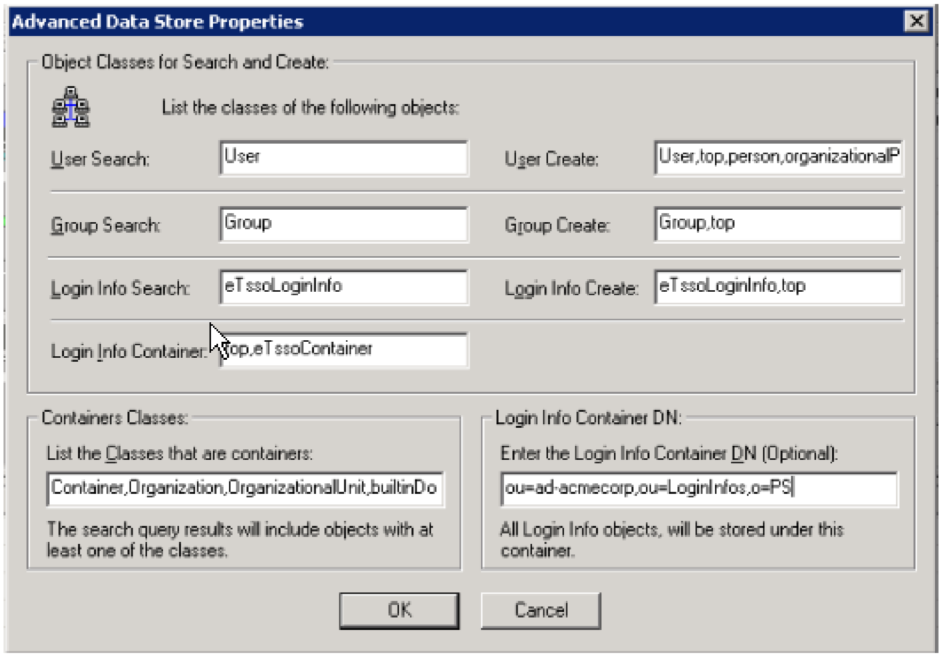
Port Number: 13389

□ SSL Connection

7. Click the Advanced button on lower right.

8. Keep all defaults except modify/add the following:

   Container Classes:
   container,organization,organizationalUnit,builtinDomain,country
   Login Info Container DN: ou=ad-acmecorp,ou=LoginInfos,o=PS



Advanced Data Store Properties

Object Classes for Search and Create:

List the classes of the following objects:

User Search: User

User Create: User,top,person,organizationalP

Group Search: Group

Group Create: Group,top

Login Info Search: eTssoLoginInfo

Login Info Create: eTssoLoginInfo,top

Login Info Container: top,eTssoContainer

Containers Classes:

List the Classes that are containers: Container,Organization,OrganizationalUnit,builtinDo

The search query results will include objects with at least one of the classes.

Login Info Container DN:

Enter the Login Info Container DN (Optional): ou=ad-acmecorp,ou=LoginInfos,o=PS

All Login Info objects, will be stored under this container.

   Note: You must remove the angle brackets "<" and ">" that may appear in the LoginInfoContainerDN field - these are only here to indicate that you must enter text.

   Note: The Containers Classes field determine which classes the Policy Manager interprets as containers. Any typos cause problems or some containers may not appear in the user data store when viewed with Policy Manager.

9. Click OK twice to create the user data store.

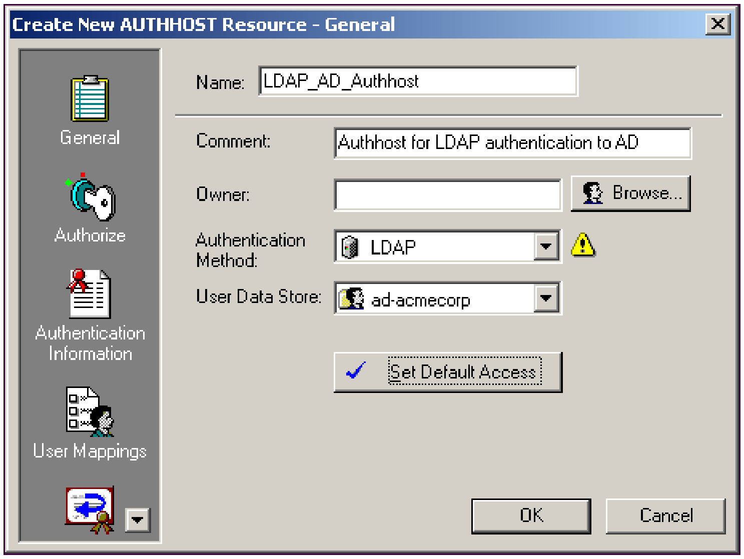
10. When asked, restart the CA SSO Server service.

Step 5d: Create a New LDAP Authentication Host

Create a new LDAP authentication host to define which users can use LDAP authentication with this user data store.

To create a new authentication host

1. Log in to Policy Manager.
2. Go to Resources, Single Sign-On Resources, Configuration Resources, Authentication Host.
3. Right-click in the right pane and select New from the menu.
4. Enter the following in the dialog box:
   - Name: LDAP_AD_Authhost
   - Comment: Authhost for LDAP authentication to AD
   - Owner: [blank]
   - Authentication Method: LDAP
   - User Data Store: ad-acmecorp



Create New AUTHHOST Resource - General

Name: LDAP_AD_Authhost

Comment: Authhost for LDAP authentication to AD

Owner:

Authentication Method: LDAP

User Data Store: ad-acmecorp

You must add users or user groups who can authenticate to Active Directory.

5. Click on the Authorize icon in the left pane.

   The Create New Authhost Resource – Authorize dialog appears.

   To authorize a user to authenticate to the LDAP Active Directory authhost:

   a. Click the + button.
      The Add Access Control List Accessor dialog appears.

   b. Select the datastore = ad-acmecorp, select user, select Browse, select All.

      The User Selection dialog appears.

   c. Browse for Philippe Perron in Human Resources, click Add, click OK.

      Philippe Perron can now authenticate to LDAP AD.

   d. Repeat steps 5a to 5c add Prani Patil (Help Desk) and Penelope Price (Reception).

      Alternatively, authorize a group of users; each user in the group can authenticate to the LDAP Active Directory authhost.

6. Click on the Authentication Information icon in the left pane

   a. Enter the information as shown:

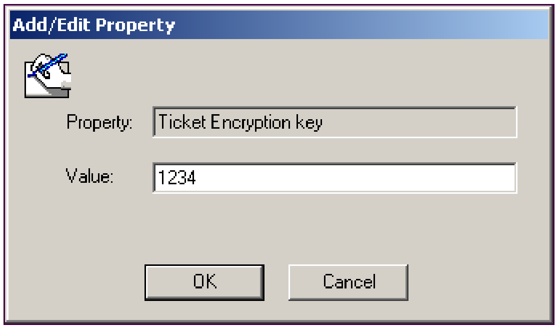
      - Name: LDAP_AD_Authhost (automatic)
      - Provider = AD
      - Authentication Data Store = ad-acmecorp

   b. Click Advanced Authentication Information

      The Advanced Authentication Information dialog appears.

   c. Double-click Ticket Encryption Key

      The Add/Edit Property dialog appears.



Add/Edit Property

Property: Ticket Encryption key

Value: 1234

d. Enter an Encryption Key value.

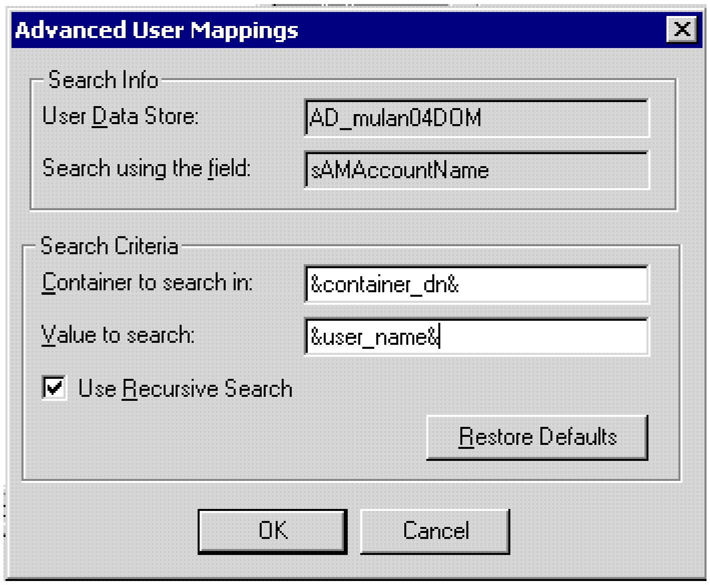
Keep a note of this value. This value must match the encryption key value that you enter when you install the LDAP authentication agent. Ticket Encryption keys have a maximum length of 256 characters

The encryption key is used by the auth agents to encrypt the SSO ticket. The CA SSO Server must have this value to decrypt the SSO ticket during authentication.

7. Click on the User Mappings icon in the left pane

a. Select the Advanced User Mappings button.

Make sure the user mapping information is the same as the following screen:



Advanced User Mappings

Search Info

User Data Store: AD_mulan04DOM

Search using the field: sAMAccountName

Search Criteria

Container to search in: &container_dn&

Value to search: &user_name&

☑ Use Recursive Search

8. Click OK twice to save your settings.

9. Exit the Policy Manager.

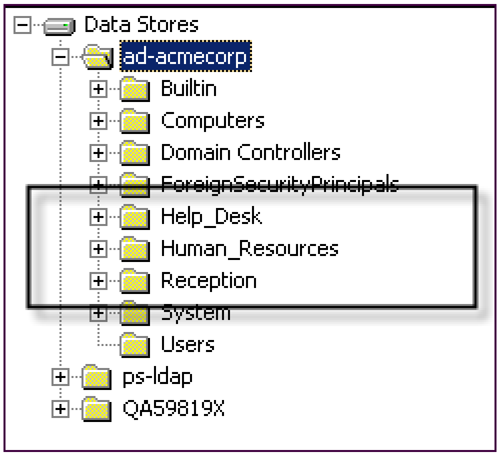
Step 5e: Verify User Data Store Configuration

To verify the User Data Store Configuration

1. Restart the “CA CA SSO Server” service.

2. Log in to Policy Manager.

3. Go to Users. Expand the “ad-acmecorp” data store and select the Users container. The AD users and groups should be displayed.



4. Select the Human_Resources folder.

   The View or Set User Properties – General dialog appears.

Check that Philippe Perron is listed. If you had linked Philippe to a group, you can click the Groups icon in the left pane to see which groups he was linked to.

Note: The Policy Manager can only be used to view users or read attributes in Active Directory. To create or modify users, you should use the AD tools.

Step 5f: Authorize SSO Resources to Active Directory User Groups

Various SSO resources need to be linked to the SSO-specific users or groups in AD. This authorizes AD users to access appropriate authentication methods, authentication host groups, application groups, and session profiles.

To Authorize SSO Resources to Active Directory User Groups

1. Log onto the Policy Manager.
2. Go to Resources, Single Sign-on Resources, Configuration Resources, Authentication Host Group
3. Right-click in the right pane and select New.

   The Create New GAUTHHOST Resource – General dialog appears.

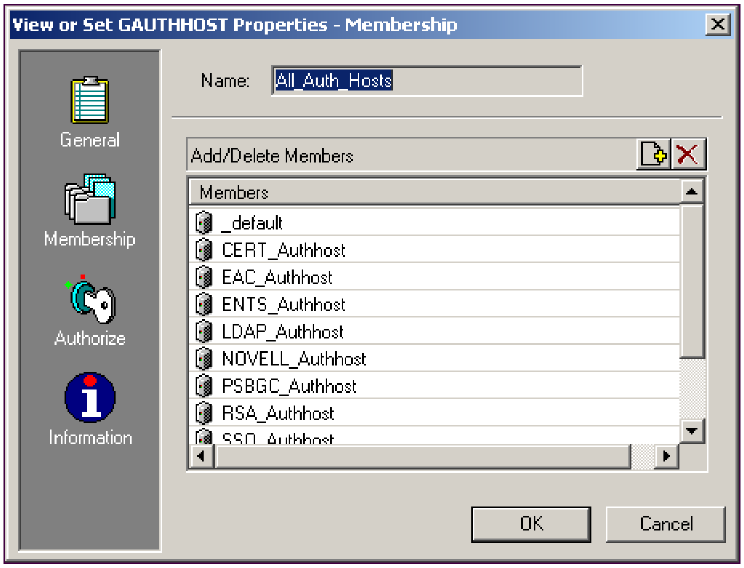
   Enter the name All_Auth_Host.

Name: All_Auth_Host

4. Click the Membership icon in the left pane.

   The Create New GAUTHHOST Resource – Membership dialog appears.

5. Add all the existing Auth Hosts, one by one, using the "+" button. You can select multiple hosts at once using either the Shift or Ctrl key. It works the same way as selecting multiple files in Windows Explorer.



View or Set GAUTHHOST Properties - Membership

Name: All_Auth_Hosts

Add/Delete Members

Members

_default
CERT_Authhost
EAC_Authhost
ENTS_Authhost
LDAP_Authhost
NOVELL_Authhost
PSBGC_Authhost
RSA_Authhost
SSO_Authhost

6. Click the Authorize icon in the left pane

   The View or Set GAUTHHOST Properties – Authorize dialog appears.

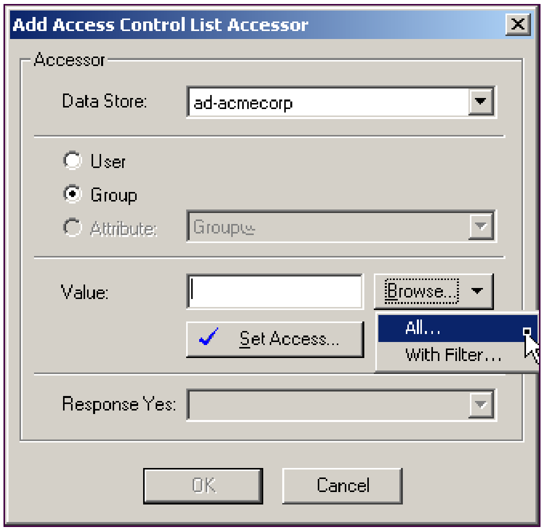
7. Click the "+" button.

   The Add Access Control List Accessor dialog appears.

8. From the Data Store drop down, select ad-acmecorp

   Select the Group radio button

   Click Browse, All



Add Access Control List Accessor

Accessor

Data Store: ad-acmecorp

○ User
● Group
○ Attribute: Groups

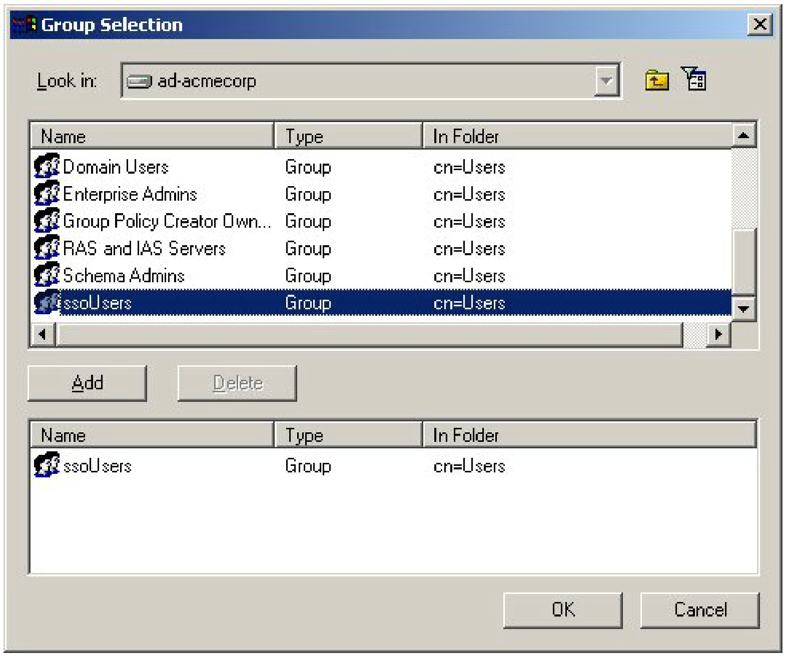
Value:

   The Group Selection dialog appears.

9. Select the Users folder.

   All Groups in the Users folder appear in the top pane.

10. Select ssoUsers and click Add.



Group Selection

Look in: ad-acmecorp

Name	Type	In Folder
Domain Users	Group	cn=Users
Enterprise Admins	Group	cn=Users
Group Policy Creator Own...	Group	cn=Users
RAS and IAS Servers	Group	cn=Users
Schema Admins	Group	cn=Users
ssoUsers	Group	cn=Users

Name	Type	In Folder
ssoUsers	Group	cn=Users

11. Click OK three times to save.

Step 5g: Apply Resources

To apply resources

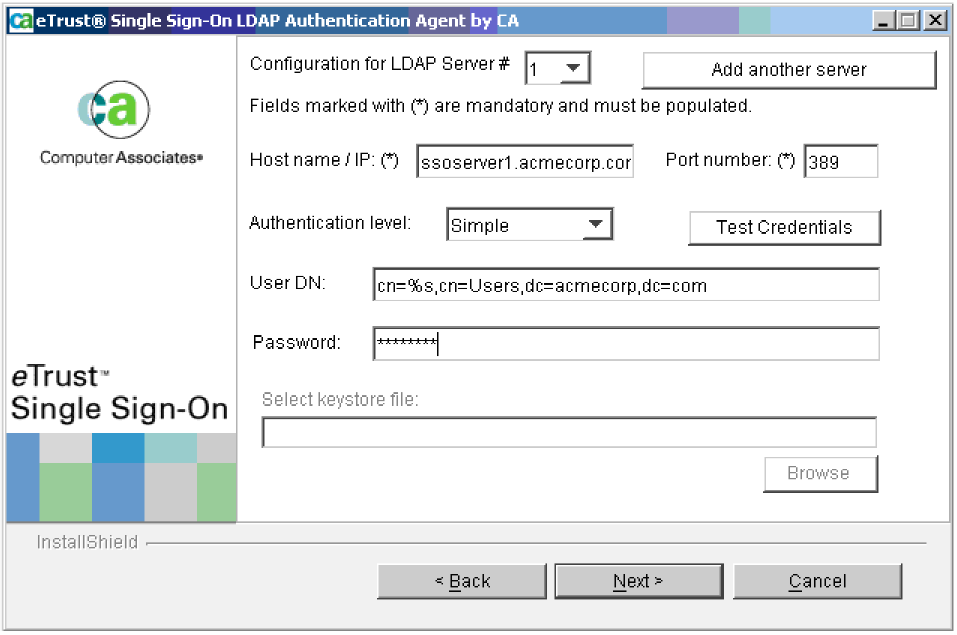
1. Go to Resources, Single Sign-On Resources, Application Resources, Applications.
2. Create an application.
   a. Right-click in the right pane and select New.
   b. Enter “Trial Apps” as the Name.
   c. Click OK.
3. Similar to Authentication Host Groups, authorize the SSO specific user groups in AD to the appropriate applications.
4. Go to Resources, Single Sign-On Resources, Configuration Resources, Authentication Method.
5. Similar to Authentication Host Groups, authorize the SSO-specific user groups in AD to the appropriate authentication methods. For now, authorize WIN and LDAP authentication methods.

   Note: You can now link authentication methods to user groups as well. This feature was added since the AD user record doesn’t know anything about authentication methods.
6. Similar to Authentication Host Groups, authorize the SSO-specific user groups in AD to the appropriate Session Profiles. Skip this step if Session Management isn’t going to be used.

Step 6: Install and Configure LDAP Authentication Agent

To Install and configure the LDAP authentication agent

1. On SSOAuthAg.AcmeCorp.com (this must be a different machine from where you have configured Active Directory), commence an installation of the SSO LDAP Authentication Agent.



eTrust® Single Sign-On LDAP Authentication Agent by CA

Configuration for LDAP Server # 1

Fields marked with (*) are mandatory and must be populated.

Host name / IP: (*) ssoserver1.acmecorp.cor

Port number: (*) 389

Authentication level: Simple

User DN: cn=%s,cn=Users,dc=acmecorp,dc=com

Password: ********

Select keystore file:

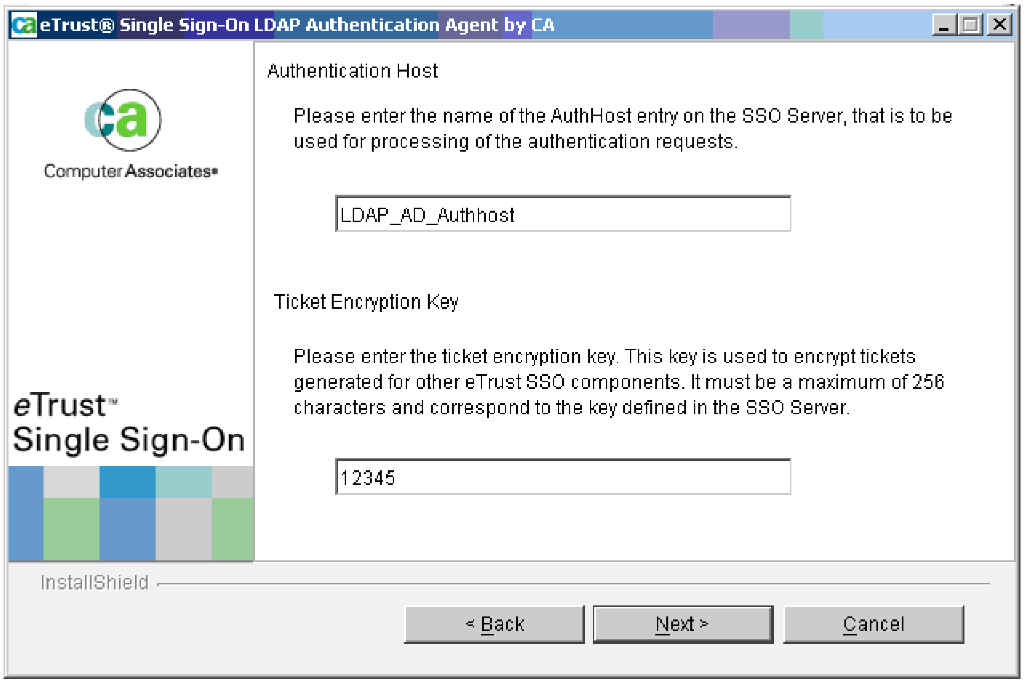
2. In the 'Authentication' dialog, enter the name of the computer where the Active Directory service was installed, port number on which Active Directory is listening for communication queries (389 is the default), and cn=Prani Patil, ou= Help Desk, dc=acmecorp, dc=com in the last field.

   You can test the credentials you have entered using the 'Test Credentials' button.

3. Enter the name mapping you want the auth agent to use when searching Active Directory for the user's credentials:

   - Base search DN: dc=AcmeCorp,dc=com
   - Search scope type: leave as 'Subtree'
   - Search filter: sAMAccountName=%s

4. When prompted in the following dialog, enter the value of the Auth Host that was created before. You must also enter the key value you want to use to encrypt tickets created by the LDAP Auth Agent.



Authentication Host

Please enter the name of the AuthHost entry on the SSO Server, that is to be used for processing of the authentication requests.

LDAP_AD_Authhost

Ticket Encryption Key

Please enter the ticket encryption key. This key is used to encrypt tickets generated for other eTrust SSO components. It must be a maximum of 256 characters and correspond to the key defined in the SSO Server.

12345

This key value must match the Key field value defined for the LDAP_AD_Authhost Auth Host entry in the Policy Manager.

5. Review the summary information, and then click Install.

6. Select Start, Run, enter ‘notepad’ in the text field and press OK.

7. Check the %ProgramFiles%\CA\Single Sign-On\LDAP Agent\LDAPAgent.log file to ensure you do not see any error message. If so, please consult the Troubleshooting section of this document. If not, the LDAPAgent.log file should look something like the following:

```
################################################################################
#
# Created Appender on: 02-15-04 22:47:15
#
################################################################################
2004-02-16 09:47:15 INFO tga_ldap [] - File version: 325,0,0,0
2004-02-16 09:47:15 INFO tga_ldap [] - Product version: 7,0,0,0
2004-02-16 09:47:15 INFO tga_ldap [] - Build description: Beta Build; Build date: Fri Jan 23 01:10:28
AUSEDT 2004
2004-02-16 09:47:16 INFO tga_ldap [] - CA SSO - LDAP Authentication Agent - Agent1 is installed.
 ################################################################################
#
# Created Appender on: 02-15-04 22:47:25
#
################################################################################
2004-02-16 09:47:25 INFO tga_ldap [] - File version: 325,0,0,0
2004-02-16 09:47:25 INFO tga_ldap [] - Product version: 7,0,0,0
2004-02-16 09:47:25 INFO tga_ldap [] - Build description: Beta Build; Build date: Fri Jan 23 01:10:28
AUSEDT 2004
2004-02-16 09:47:25 INFO tga_ldap [] - Using PortNumber 17979
2004-02-16 09:47:25 INFO tga_ldap [] - ChildLimit: 3
2004-02-16 09:47:25 INFO tga_ldap [] - IdleFreq: 20
2004-02-16 09:47:25 INFO tga_ldap [] - TimeOutConnect: 60
2004-02-16 09:47:25 INFO tga_ldap [] - TimeOutRecv: 60
2004-02-16 09:47:25 INFO tga_ldap [] - TimeOutSend: 30
2004-02-16 09:47:25 INFO tga_ldap [] - SendBuffSize: 131072
2004-02-16 09:47:25 INFO tga_ldap [] - RecvBuffSize: 131072
2004-02-16 09:47:25 INFO tga_ldap [] - TicketKey: 1234
2004-02-16 09:47:25 INFO tga_ldap [] - PolicyFilePath: D:\Program Files\CA\Sign Sign-On\LDAP
Agent\tga_ldapPolicy.ini
2004-02-16 09:47:25 INFO tga_ldap [] - AuthHostName: LDAP_ps-ldap
2004-02-16 09:47:25 INFO tga_ldap [] - UserNamePrefix:
2004-02-16 09:47:25 INFO tga_ldap [] - UserNameSuffix:
2004-02-16 09:47:25 INFO tga_ldap [] - StandbyConnections: 5
2004-02-16 09:47:25 INFO tga_ldap [] - MaxConnections: 10
2004-02-16 09:47:25 INFO tga_ldap [] - SearchTimeout: 120
2004-02-16 09:47:25 INFO tga_ldap [] - OfflineTimeout: 120
2004-02-16 09:47:25 INFO tga_ldap [] - ConnectionLifetime: 3600
2004-02-16 09:48:51 INFO tga_ldap [] - service_ctrl calling ServiceStop
```

```
##########################################################################################
#
# Created Appender on: 02-15-04 22:48:52
#
##########################################################################################
2004-02-16 09:48:52 INFO tga_ldap [] - File version: 325,0,0,0
2004-02-16 09:48:52 INFO tga_ldap [] - Product version: 7,0,0,0
2004-02-16 09:48:52 INFO tga_ldap [] - Build description: Beta Build; Build date: Fri Jan 23 01:10:28
AUSEDT 2004
2004-02-16 09:48:52 INFO tga_ldap [] - Using PortNumber 17979
2004-02-16 09:48:52 INFO tga_ldap [] - ChildLimit: 3
2004-02-16 09:48:52 INFO tga_ldap [] - IdleFreq: 20
2004-02-16 09:48:52 INFO tga_ldap [] - TimeOutConnect: 60
2004-02-16 09:48:52 INFO tga_ldap [] - TimeOutRecv: 60
2004-02-16 09:48:52 INFO tga_ldap [] - TimeOutSend: 30
2004-02-16 09:48:52 INFO tga_ldap [] - SendBuffSize: 131072
2004-02-16 09:48:52 INFO tga_ldap [] - RecvBuffSize: 131072
2004-02-16 09:48:52 INFO tga_ldap [] - TicketKey: 1234
2004-02-16 09:48:52 INFO tga_ldap [] - PolicyFilePath: D:\Program Files\CA\Single Sign-On\LDAP
Agent\tga_ldapPolicy.ini
2004-02-16 09:48:52 INFO tga_ldap [] - AuthHostName: LDAP_ps-ldap
2004-02-16 09:48:52 INFO tga_ldap [] - UserNamePrefix:
2004-02-16 09:48:52 INFO tga_ldap [] - UserNameSuffix:
2004-02-16 09:48:52 INFO tga_ldap [] - StandbyConnections: 5
2004-02-16 09:48:52 INFO tga_ldap [] - MaxConnections: 10
2004-02-16 09:48:52 INFO tga_ldap [] - SearchTimeout: 120
2004-02-16 09:48:52 INFO tga_ldap [] - OfflineTimeout: 120
2004-02-16 09:48:52 INFO tga_ldap [] - ConnectionLifetime: 3600
```

Step 7: Install and Configure the CA SSO Client

This procedure describes how to install the CA SSO Client using the Product Explorer.

To install the CA SSO Client

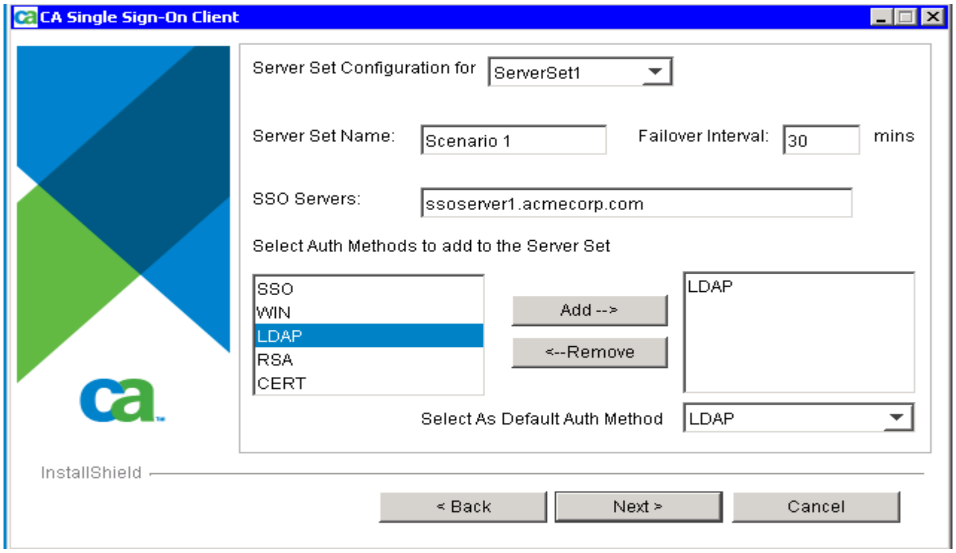
1. Insert the product installation DVD into your DVD-ROM drive.

   If you have autorun enabled, the Product Explorer automatically displays. Otherwise, navigate to the CD-ROM drive and double-click the PE_i386.EXE file.

2. From the Product Explorer menu, select CA SSO Client, and click Install.
3. Select only the SSO and LDAP Authentication Methods to be installed.

4. On the Server Set Configuration dialog, create a server set using the following information:

Server Set Name:	Scenario 1
CA SSO Servers:	ssoserver1.acmecorp.com
Failover Interval:	30 minutes
Auth Methods:	LDAP


CA Single Sign-On Client

Server Set Configuration for: ServerSet1

Server Set Name: Scenario 1

Failover Interval: 30 mins

SSO Servers: ssoserver1.acmecorp.com

Select Auth Methods to add to the Server Set

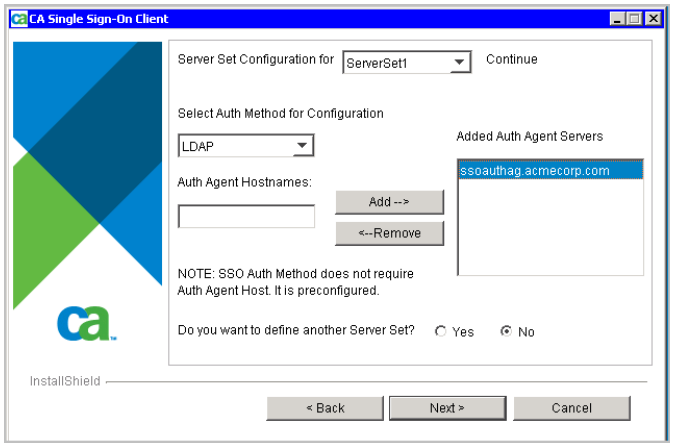
- SSO
- WIN
- LDAP
- RSA
- CERT

LDAP

Select As Default Auth Method: LDAP

5. On the Authhost configuration dialog configure your authhost using the following information:

- Auth Method: LDAP
- Auth Agent Server: ssoauthag.acmecorp.com


CA Single Sign-On Client

Server Set Configuration for: ServerSet1 Continue

Select Auth Method for Configuration: LDAP

Auth Agent Hostnames:

Added Auth Agent Servers: ssoauthag.acmecorp.com

NOTE: SSO Auth Method does not require Auth Agent Host. It is preconfigured.

Do you want to define another Server Set? ○ Yes ● No

If you wish to add another server set, select Yes, otherwise click Next.

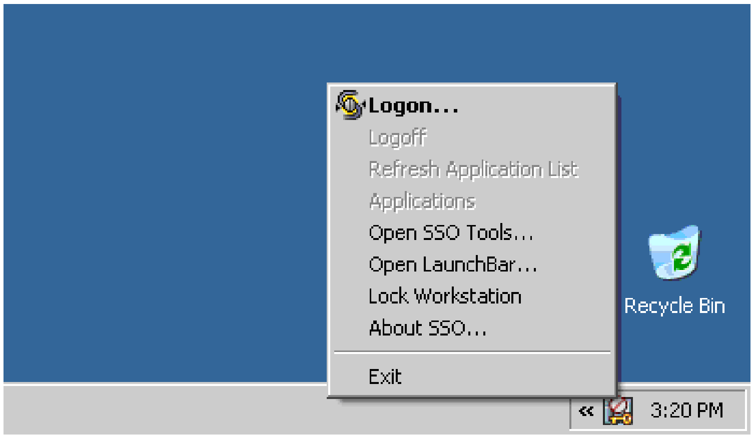
6. On the Summary dialog, check the information.
7. Click Install.

   Note: When the message InstallShield Wizard Complete appears, you have successfully installed the CA SSO Client. Make sure to reboot your computer when the prompt appears.

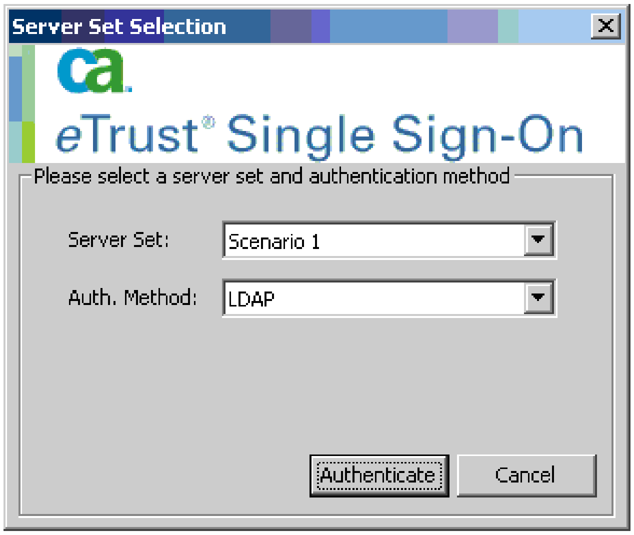
Step 8: Authenticate to Active Directory from the CA SSO Client

To authenticate to Active Directory from the CA SSO Client

1. After finishing the CA SSO Client install and rebooting ssoclient.acmecorp.com, the SSOStatus icon appears in the system tray.
2. Right-click the icon and select Logon:



The server set selection wizard appears:


Server Set Selection

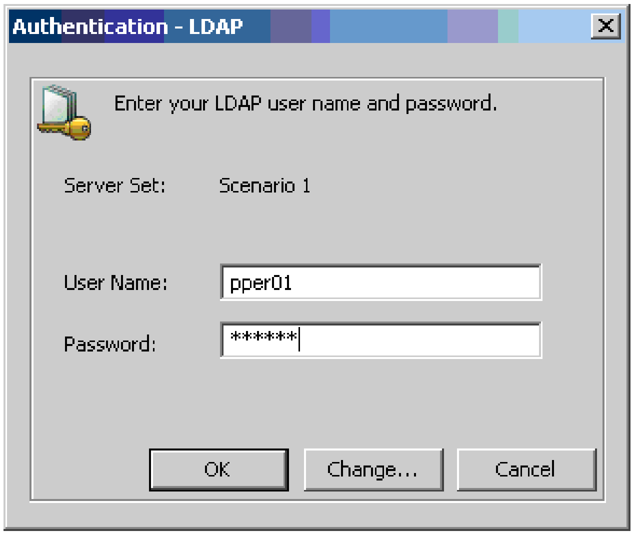
eTrust® Single Sign-On

Please select a server set and authentication method

Server Set: Scenario 1

Auth. Method: LDAP

3. Click Authenticate. The LDAP authentication dialog appears:



Server Set: Scenario 1

User Name: pper01

Password: ******

4. Enter 'pper01' in the user name field and the appropriate password, then click OK.

   The CA SSO Client authenticates and logs Philippe on to the CA SSO Server using LDAP authentication.

   The SSO Status Icon changes to a green tick in the tray menu on the bottom right-hand corner of the screen. Hovering the mouse over the icon will display a tool tip showing that the logged on SSO user is Philippe Perron.

   The user can right-click this icon and select Applications from the menu to access their list of SSO-enabled applications. Alternatively, the user can right-click the SSO Status Icon and select one of SSO Tools or the SSO Launchbar to view their applications.

Step 9: Create and Test an Application

This section gives you a basic Tcl script that can be used to launch an application from the SSO Launchbar.

Step 9a: Create a Logon Script

Here is a logon script that launches Notepad and types the name of the user currently logged in.

```
sso run -path "notepad.exe"
sso window -titleglob "*Notepad*"
sso type -text "Logged in as user $_USERNAME"
```

Create a file named note.tcl at C:\Program Files\CA\Single Sign-On\Server\Scripts\ that contains the above example.

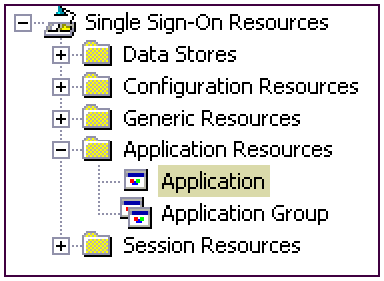
For more information about writing Tcl scripts to log users in and out of applications and documents, see the *Tcl Scripting Reference Guide.*

Step 9b: Define Logon Script to the CA SSO Server

To define a Logon Script on the CA SSO Server

This procedure tells you how to define an application on the CA SSO Server.

1. Launch the Policy Manager
2. In the Program Bar navigate to Single Sign-On Resources, Application Resources, Application.

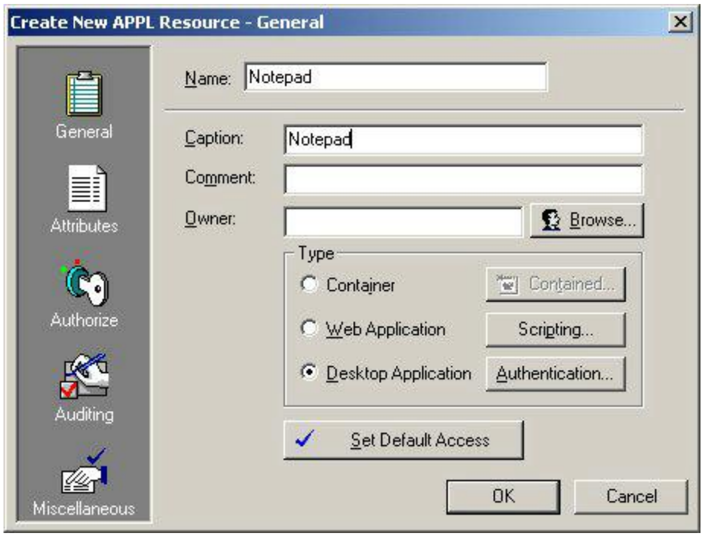
   Right-click in the Application Window and choose New.

   The Create New APPL Resource – General dialog appears.
3. Fill in the details of the application.

   For example:

   Name: Notepad
   Caption: Notepad
   Type: Desktop Application

Note: The caption is what the user sees in their CA SSO Application List.



Create New APPL Resource - General

Name: Notepad

Caption: Notepad

Comment:

Owner:

Type

○ Container

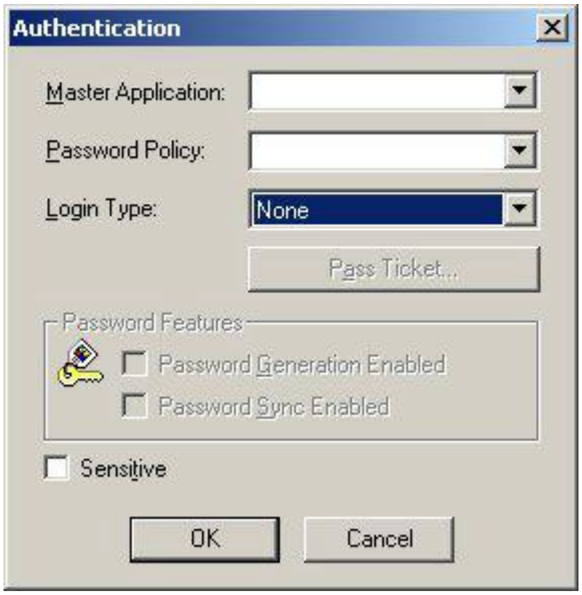
○ Web Application

● Desktop Application

4. Click the Scripting button.
   The Scripting dialog appears.

5. Enter note.tcl in the Script File field, and then click OK.

   Select the Authentication button and set the Login Type to None.



Authentication

Master Application:

Password Policy:

Login Type: None

Password Features

□ Password Generation Enabled

□ Password Sync Enabled

□ Sensitive

6. Select the Authorize icon.
   The Create New APPL Recourse – Authorize dialog appears.

7. Right-click and choose Add.
   The Add Access Control List Accessor dialog appears.

8. Add the ssoUsers group to the authorized list.

Step 9c: Launch the Application

This procedure tells you how to test the script.   This is the procedure that end-users would follow.

To launch the application

1. Using the Philippe Perron user, logon and authenticate to SSO.
   This means that you will have a current SSO ticket.
2. Choose the Notepad application from the list of SSO-enabled applications.

Step 10: Test the LDAP/AD Password Change Functionality

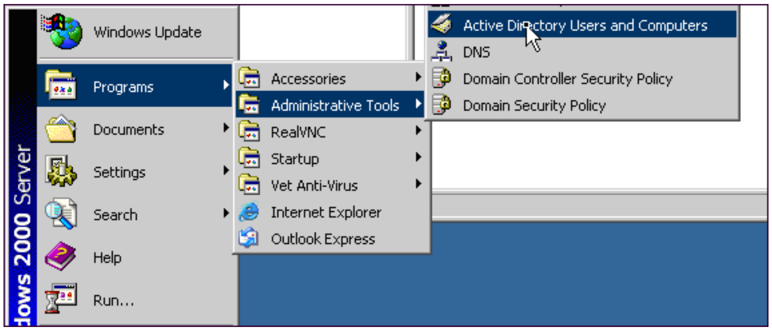
To test the LDAP/AD Password change functionality, you must:

- Set Philipe Peron's domain password to expire at next login
- Login on the Client with User Philipe Perron

Step 10a: Set Philipe Peron's Domain Password to Expire At Next Login

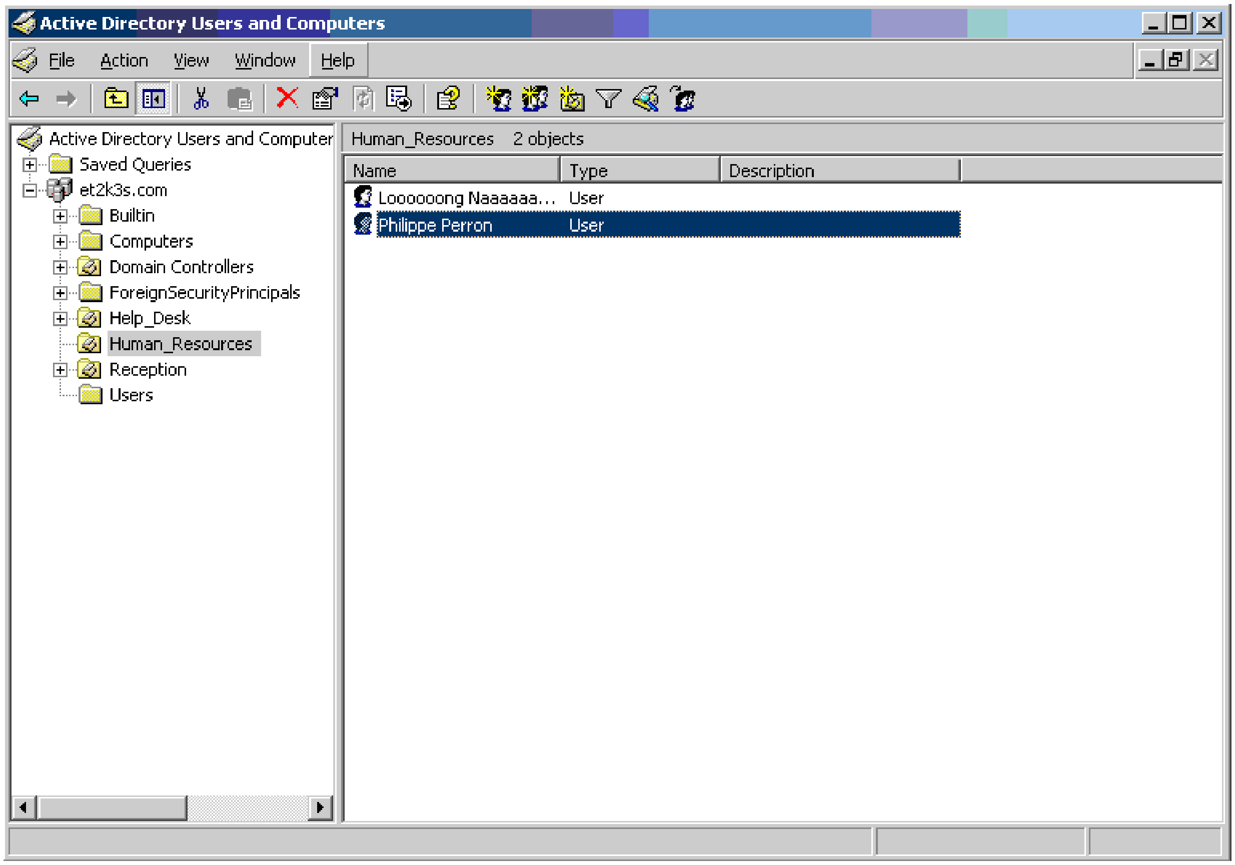
This procedure is done on the user Philippe Perron, using the Active Directory User Management snap-in.

1. Logon to SSOserver1.AcmeCorp.com as a Windows user with administrative privileges (i.e. is a member of the 'Administrators' group), preferably as a built-in 'Administrator' account.
2. From the Start menu select, Programs, Administrative Tools, Active Directory Users and Computers.



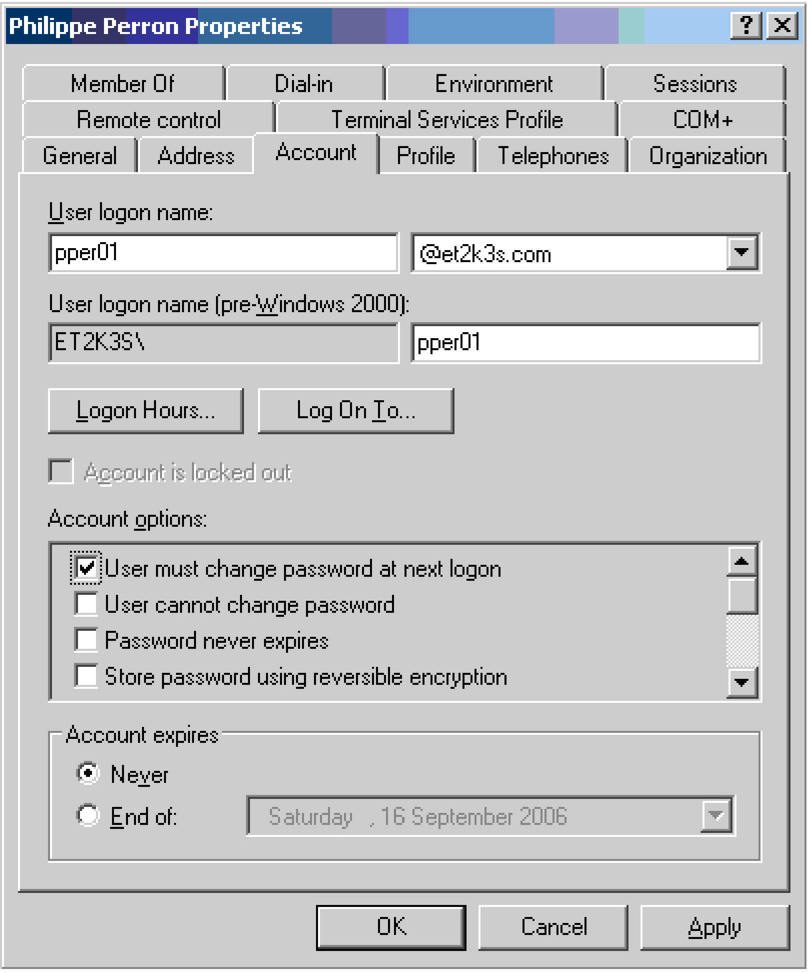
The Active Directory Users and Computers dialog appears.

3. Select the Human_Resources folder from the tree in the left pane, then right-click in the right pane and select the user Philippe Perron.


Active Directory Users and Computers
File
Action
View
Window
Help
Active Directory Users and Computer
Saved Queries
et2k3s.com
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Help_Desk
Human_Resources
Reception
Users
Human_Resources 2 objects

Name	Type	Description
Loooooong Naaaaaa...	User	
Philippe Perron	User	

The User Properties dialog for Philippe Peron appears.



User logon name: pper01 @et2k3s.com

User logon name (pre-Windows 2000): ET2K3S\ pper01

□ Account is locked out

Account options:

☑ User must change password at next logon
□ User cannot change password
□ Password never expires
□ Store password using reversible encryption

Account expires

● Never
○ End of: Saturday , 16 September 2006

4. Select the Accounts tab and check the box for User must change password at next logon.
5. Save this change.

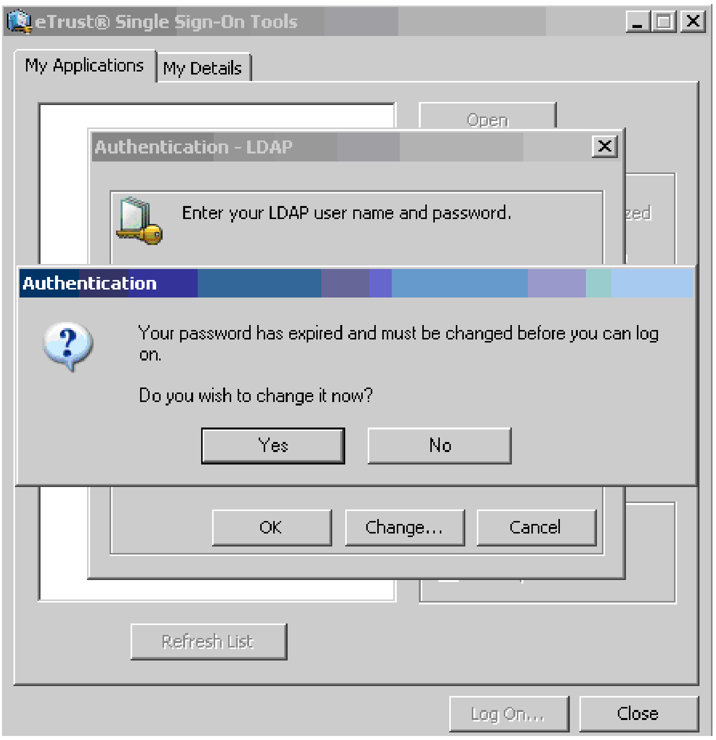
Step 10b: Login on the Client with User Philipe Perron

On the SSO Client machine, log off (if the user is still logged on).

1. Open CA SSO Tools:
2. Right-click on the SSOStatus icon and select Logon
3. Select the server set and click Authenticate.

4. Enter pper01 and Philippe's LDAP (Active Directory) password and select Ok.

   The prompt to change domain password appears:



Enter your LDAP user name and password.

Your password has expired and must be changed before you can log on.

Do you wish to change it now?

5. Select Yes and enter a new password for Philippe Peron.

   The new password is accepted and Philippe's domain password is changed.